Thai Journal of Mathematics
Volume 14 (2016) Number 3 : 599–613

http://thaijmath.in.cmu.ac.th
ISSN 1686-0209

Numerical Solution of Convection-Diffusion Equation Using Cubic B-Spline Quasi-Interpolation

Hossein Aminikhah[1] and Javad Alavi

Department of Applied Mathematics, School of Mathematical Sciences
University of Guilan, P.O. Box 1914, P.C. 41938, Rasht, Iran
e-mail : aminikhah@guilan.ac.ir (H. Aminikhah)
javadealavi@gmail.com (J. Alavi)

Abstract : In this paper, the convection-diffusion equation with Dirichlet's type boundary conditions is solved numerically by cubic B-spline quasi-interpolation. The numerical scheme, obtained by using the derivative of the quasi-interpolation to approximate the spatial derivative of the dependent variable and first order forward difference to approximate the time derivative of the dependent variable. The developed method is tested on various problems and the numerical results are reported in tabular and graphical form. Easy and economical implementation process is the strength of the scheme. The results of numerical experiments are compared with analytical solutions by calculating errors L_2, L_∞-norms.

Keywords : B-spline; quasi-interpolation; convection-diffusion equation; difference schemes.
2010 Mathematics Subject Classification : 41A15; 65N06.

1 Introduction

The term convection means the movement of molecules within fluids, whereas, diffusion describes the spread of particles through random motion from regions of higher concentration to regions of lower concentration. Convection-diffusion equations model a variety of physical phenomena. The numerical solution of convection-

[1]Corresponding author.

diffusion transport problems arises in many important applications in science and engineering. Characteristic examples are the heat transfer through a permeable medium, the transport of a pollutant through the atmosphere or the transport of a fluid through the porous medium. Various numerical techniques have been developed and compared for solving the one dimensional convection-diffusion equation with constant coefficient [1, 2, 3, 4, 5, 6]. The piecewise polynomial, especially B-spline, has become a fundamental tool for numerical methods to get the solution of the differential equations [7, 8, 9, 10, 11]. The numerical solutions of partial differential equations by B-spline quasi-interpolation were introduced in [12, 13, 14]. B-splines in the collocation and Galerkin methods are introduced for the numerical solutions of the convection-diffusion equation in [15, 16]. In this paper, we provide a numerical scheme to solve convection-diffusion equation using the derivative of the cubic B-spline quasi-interpolation to approximate the spatial derivative of the differential equations and utilize first order forward difference to approximate the time derivative. The convection-diffusion equation is given by

$$u_t + \epsilon u_x = \gamma u_{xx}, \quad 0 < x < L, \quad 0 < t \leq T \tag{1.1}$$

with initial condition

$$u(x,0) = \phi(x) \tag{1.2}$$

and boundary conditions are of the form

$$u(0,t) = g_0(t), \quad u(1,t) = g_1(t), \quad t \in [0,T] \tag{1.3}$$

where the parameters γ, $\epsilon > 0$ are the viscosity coefficient and phase speed respectively and subscripts t and x denote differentiation. g_0, g_1 and ϕ are known functions with sufficient smoothness. This paper is organized as follows. In Section 2, we obtain the numerical schemes using cubic B-spline quasi-interpolation to solve convection-diffusion equation (1.1). Numerical experiments for various test problems are solved to assess the accuracy of the technique and the maximum absolute errors will be presented in Section 3. Finally, we give some concluding remarks in Section 4.

2 Numerical Scheme Using Cubic B-Spline Quasi-Interpolant

Given a bounded interval $I = [0, L]$, denoted by $S_d(X_n)$ the space of splines of degree d and class C^{d-1} on the uniform partition $X_n = \{x_i = ih, i = 0, 1, \ldots, n\}$ with meshlength $h = L/n$. Let a basis of $S_d(X_n)$ be $\{B_{j,d,r},\ h = 1, 2, \ldots, n + d\}$ where $B_{j,d,r}$ is the j th B-spline of degree d for the knot sequence $r := (r_i)_{i=-d}^{n+d}$, $r_{-d} = r_{-d+1} = \ldots = r_{-1} = 0,\ r_i = x_i,\ 0 \leq i \leq n$ and $r_n = r_{n+1} = \ldots = r_{n+d} = L$. Since the cubic spline has become the most commonly used spline we use cubic B-spline quasi-interpolation in this paper. Discretizing Eq (1.1) in time, we get

$$u_i^{k+1} = u_i^k + \tau(\gamma(u_{xx})_i^k - \epsilon(u_x)_i^k) \tag{2.1}$$

where u_i^k is the approximation of the value $u(x,t)$ at (x_i,t_k), $t_k = k\tau$, and τ is the time step. Then for fixed k, we can get the cubic quasi-interpolation as follows [17]:

$$Q_3u^k = \sum_{j=1}^{n+3} \mu_j(u^k)B_{j,3,r}(x) \tag{2.2}$$

where $u^k = u(x,t_k)$ and the coefficient functionals are:

$$\begin{aligned}
\mu_1(u^k) &= u_0^k \\
\mu_2(u^k) &= \frac{1}{18}(7u_0^k + 18u_1^k - 9u_2^k + 2u_3^k) \\
\mu_j(u^k) &= \frac{1}{6}(-u_{j-3}^k + 8u_{j-2}^k - u_{j-1}^k)\ ,\ 3 \le j \le n+1 \\
\mu_{n+2}(u^k) &= \frac{1}{18}(2u_{n-3}^k - 9u_{n-2}^k + 18u_{n-1}^k + 7u_n^k) \\
\mu_{n+3}(u^k) &= u_n^k.
\end{aligned}$$

From [17], we have the error estimate

$$||U^k - Q_3u^k||_\infty = O(h^4). \tag{2.3}$$

For approximate derivatives of u^k by derivatives of Q_3u^k up to the order h^3 at x_i we can evaluate the value of u^k at x_i by:

$$(Q_3u_i^k)^{'} = \sum_{j=1}^{n+3} \mu_j(u^k)B_j^{'}(x_i),\ (Q_3u_i^k)^{''} = \sum_{j=1}^{n+3} \mu_j(u^k)B_j^{''}(x_i).$$

We set

$$\begin{aligned}
U^k &= (u_0^k, u_1^k, \ldots, u_n^k)^T, \\
U_x^k &= ((u_0^k)^{'}, (u_1^k)^{'}, \ldots, (u_n^k)^{'}), \\
U_{xx}^k &= ((u_0^k)^{''}, (u_1^k)^{''}, \ldots, (u_n^k)^{''}),
\end{aligned}$$

where

$$(u_i^k)^{'} = (Q_3u_i^k)^{'},\ (u_i^k)^{''} = (Q_3u_i^k)^{''},\quad i = 0,1,\ldots,n.$$

By solution of the linear systems

$$\begin{aligned}
(u_i^k)^{'} &= \sum_{j=1}^{n+3} \mu_j(u^k)B_j^{'}(x_i),\quad i = 0,1,\ldots,n \\
(u_i^k)^{''} &= \sum_{j=1}^{n+3} \mu_j(u^k)B_j^{''}(x_i),\quad i = 0,1,\ldots,n,
\end{aligned}$$

we obtain

$$U_x^k = \frac{1}{h}D_1U^k,\ U_{xx}^k = \frac{1}{h^2}D_2U^k \tag{2.4}$$

where $D_1,\ D_2 \in \mathbb{R}^{(n+1)\times(n+1)}$ are obtain as follows:

$$D_1=\begin{bmatrix} -11/6 & 3 & -3/2 & 1/3 & 0 & 0 & \dots & 0 & 0\\ -1/3 & -1/2 & 1 & -1/6 & 0 & 0 & \dots & 0 & 0\\ 1/12 & -2/3 & 0 & 2/3 & -1/12 & 0 & \dots & 0 & 0\\ 0 & 1/12 & -2/3 & 0 & 2/3 & -1/12 & \dots & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & \dots & 1/12 & -2/3 & 0 & 2/3 & -1/12 & 0\\ 0 & 0 & \dots & 0 & 1/12 & -2/3 & 0 & 2/3 & -1/12\\ 0 & 0 & \dots & 0 & 0 & 1/6 & -1 & 1/2 & 1/3\\ 0 & 0 & \dots & 0 & 0 & -1/3 & 3/2 & -3 & 11/6 \end{bmatrix}$$

$$D_2=\begin{bmatrix} 2 & -5 & 4 & -1 & 0 & 0 & \dots & 0 & 0\\ 1 & -2 & 1 & 0 & 0 & 0 & \dots & 0 & 0\\ -1/6 & 5/3 & -3 & 5/3 & -1/6 & 0 & \dots & 0 & 0\\ 0 & -1/6 & 5/3 & -3 & 5/3 & -1/6 & \dots & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & \dots & -1/6 & 5/3 & -3 & 5/3 & -1/6 & 0\\ 0 & 0 & \dots & 0 & -1/6 & 5/3 & -3 & 5/3 & -1/6\\ 0 & 0 & \dots & 0 & 0 & 0 & 1 & -2 & 1\\ 0 & 0 & \dots & 0 & 0 & -1 & 4 & -5 & 2 \end{bmatrix}.$$

From the initial conditions (1.2) and boundary conditions (1.3), we can compute the numerical solution of Eq. (1.1) step by step using the scheme (2.1) and formulas (2.4).

3 Numerical Examples

In this section we test our scheme on some examples. We tested the accuracy of this method for different values of γ and ϵ. The versatility and the accuracy of the proposed method is measured using the L_2 and L_∞ error norms for the test problems. The error norms are defined as

$$L_2=\sqrt{h\sum_{j=0}^{n}|(u_j^{exact}-u_j^{numerical})^2|}\ \ and\ \ L_\infty=\max_j|u_j^{exact}-u_j^{numerical}|.$$

Example 3.1 ([15])**.** Consider Equation (1.1) with the initial condition

$$\phi(x)=e^{\alpha x}.$$

The exact solution is given by

$$u(x,t)=e^{\alpha x+\beta t}.$$

The boundary conditions can be obtained from the exact solution. First, we take $\epsilon = 0.1,\ \gamma = 0.02,\ \alpha = 1.17712434446770,\ \beta = -0.09,\ h = 0.01,\ \tau = 0.001,\ T = 5\ and\ L = 1$. The absolute errors for some values of t and error norms are reported in Table 1. The exact solution is shown in Figure 1 and the estimated solution is shown in Figure 2. We drew absolute error function in Figure 3, to show how little its magnitude is. The exact and numerical solutions are also depicted with $\epsilon = 3.5,\ \gamma = 0.022,\ h = 0.01,\ \tau = 0.001,\ \alpha = 0.02854797991928,\ \beta = -0.0999,\ T = 5\ and\ L = 1$. in Figure 4 and Figure 5 respectively. The absolute errors for some values of t and error norms are reported in Table 2. Absolute error between the numerical and exact solution is also depicted at all mesh points in Figure 6.

Table 1: Errors of (3.1) using $\epsilon = 0.1, \gamma = 0.02, \alpha = 1.17712434446770, \beta = -0.09, h = 0.01, \tau = 0.001, T = 5, L = 1$

	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$
$x = 0.1$	2.1506e-06	2.8217e-06	3.1023e-06	3.1872e-06	3.1585e-06
$x = 0.5$	7.0601e-06	1.2276e-05	1.5562e-05	1.7383e-05	1.8176e-05
$x = 0.9$	7.6594e-06	1.1643e-05	1.4165e-05	1.5637e-05	1.6307e-05
L_2	6.4790e-07	1.0719e-06	1.3410e-06	1.4942e-06	1.5632e-06
L_∞	9.1107e-06	1.5204e-05	1.9302e-05	2.1738e-05	2.2909e-05

Table 2: Errors of (3.1) using $\epsilon = 3.5, \gamma = 0.022, \alpha = 0.02854797991928, \beta = -0.0999, h = 0.01, \tau = 0.001, T = 5, L = 1$

	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$
$x = 0.1$	1.1646e-07	1.0539e-07	9.5371e-08	8.6304e-08	7.8099e-08
$x = 0.5$	6.4136e-07	5.8039e-07	5.2521e-07	4.7527e-07	4.3009e-07
$x = 0.9$	1.1783e-06	1.0663e-06	9.6490e-07	8.7317e-07	7.9015e-07
L_2	7.5052e-08	6.7917e-08	6.1460e-08	5.5617e-08	5.0329e-08
L_∞	1.2803e-06	1.1586e-06	1.0484e-06	9.4875e-07	8.5855e-07

Example 3.2 ([15])**.** We consider the initial condition for this example as follows

$$\phi(x) = e^{\frac{-(x-2)^2}{80\gamma}}.$$

The exact solution is given by

$$u(x,t) = \sqrt{\frac{20}{20+t}} e^{\frac{-(x-2-\epsilon t)^2}{4\gamma(t+20)}}.$$

The boundary conditions can be obtained from the exact solution. In our numerical computation, we take $\epsilon = 0.8, \gamma = 0.1, h = 0.01, \tau = 0.0001, T = 5\ and\ L = 1$. The absolute errors are reported in Table 3. Approximate and exact solutions are drawn in Figure 7 and Figure 8 respectively. Absolute error for all mesh point shown in Figure 9.

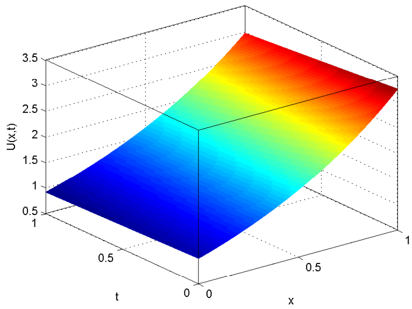

Figure 1: Exact solution of (3.1) using $\epsilon = 0.1, \gamma = 0.02, \alpha = 1.17712434446770, \beta = -0.09, h = 0.01, \tau = 0.001, T = 5, L = 1$

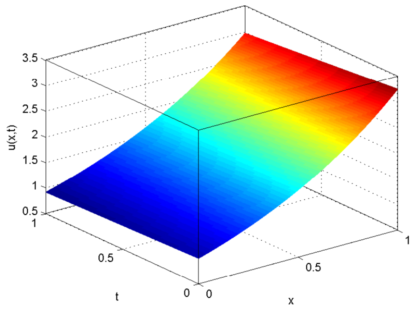

Figure 2: Numerical solution of (3.1) using $\epsilon = 0.1, \gamma = 0.02, \alpha = 1.17712434446770, \beta = -0.09, h = 0.01, \tau = 0.001, T = 5, L = 1$

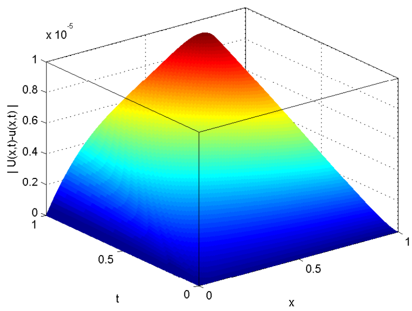

Figure 3: Absolute error of (3.1) using $\epsilon = 0.1, \gamma = 0.02, \alpha = 1.17712434446770, \beta = -0.09, h = 0.01, \tau = 0.001, T = 5, L = 1$

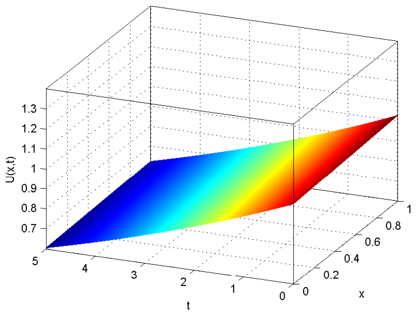

Figure 4: Exact solution of (3.1) using $\epsilon = 3.5, \gamma = 0.022, \alpha = 0.02854797991928, \beta = -0.0999, h = 0.01, \tau = 0.001, T = 5, L = 1$

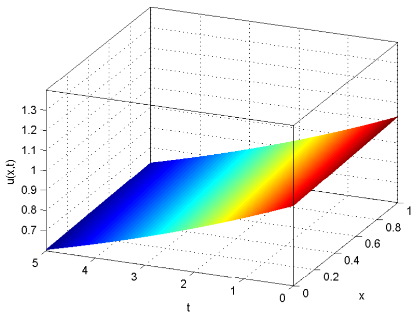

Figure 5: Numerical solution of (3.1) using $\epsilon = 3.5, \gamma = 0.022, \alpha = 0.02854797991928, \beta = -0.0999, h = 0.01, \tau = 0.001, T = 5, L = 1$

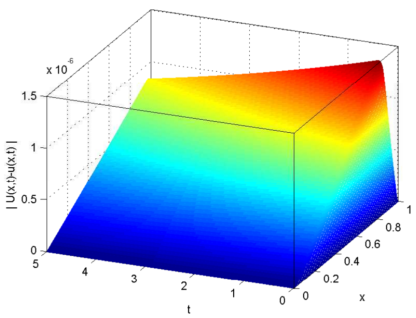

Figure 6: Absolute error of (3.1) using $\epsilon = 3.5, \gamma = 0.022, \alpha = 0.02854797991928, \beta = -0.0999, h = 0.01, \tau = 0.001, T = 5, L = 1$

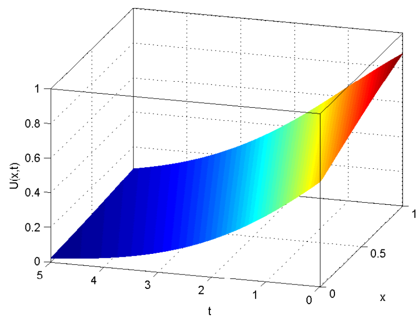

Figure 7: Exact solution of (3.2) using $\epsilon = 0.8, \gamma = 0.1, h = 0.01, \tau = 0.0001, T = 5$ *and* $L = 1$

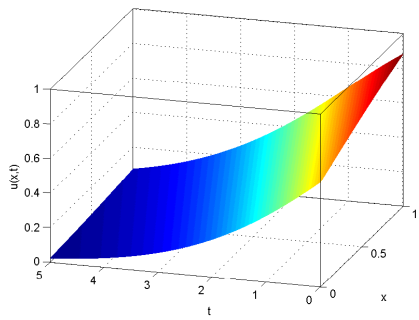

Figure 8: Numerical solution of (3.2) using $\epsilon = 0.8, \gamma = 0.1, h = 0.01, \tau = 0.0001, T = 5$ *and* $L = 1$

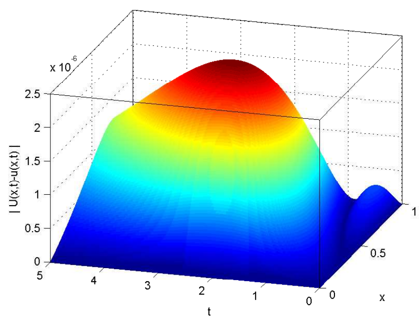

Figure 9: Absolute error of (3.2) using $\epsilon = 0.8, \gamma = 0.1, h = 0.01, \tau = 0.0001, T = 5$ *and* $L = 1$

Table 3: Errors of (3.2) using $\epsilon = 0.8, \gamma = 0.1, h = 0.01, \tau = 0.0001, T = 5$ *and* $L = 1$

	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
$x = 0.1$	2.5554e-07	3.3519e-07	2.8761e-07	1.9785e-07	1.1726e-07
$x = 0.5$	7.7590e-07	1.6315e-06	1.6606e-06	1.2651e-06	2.6567e-07
$x = 0.9$	3.0883e-07	1.4544e-06	1.7161e-06	1.4086e-06	9.4519e-07
L_2	5.6389e-08	6.1.3863e-07	1.4937e-07	1.1760e-07	7.6859e-08
L_∞	7.7622e-07	1.9618e-06	2.2098e-06	1.7807e-06	1.1819e-06

Example 3.3 ([15]). *In this example, we consider the* (1.1) *with the initial condition*

$$\phi(x) = e^{\frac{-(x-x_0)^2}{2\sigma_0{}^2}}.$$

The exact solution is given by

$$u(x,t) = \frac{\sigma_0}{\sigma} e^{\frac{-(x-x_0-\epsilon t)^2}{2\sigma_0}}$$

where $\sigma^2 = \sigma_0^2 + 2\gamma t$. *The boundary conditions can be obtained from the exact solution. First, we take* $\epsilon = 1,\ \gamma = 0.01,\ x_0 = -05,\ \sigma_0 = 0.025,\ h = 0.01,\ \tau = 0.001,\ T = 2$ *and* $L = 1$. *The absolute errors for some values of* t *and error norms are reported in Table 4. The exact and numerical solution is shown in Figure 10 and Figure 11, respectively. The error function shown in Figure 12. The exact and numerical solutions are also depicted with* $\epsilon = 0.5,\ \gamma = 0.01,\ x_0 = 1,\ \sigma_0 = 025,\ h = 0.01,\ \tau = 0.001,\ T = 6,\ L = 4$ *in Figure 13 and Figure 14 respectively. The absolute errors for some values of and error norms are reported in Table 5. Absolute error between the numerical and exact solution is also depicted at all mesh points in Figure 15.*

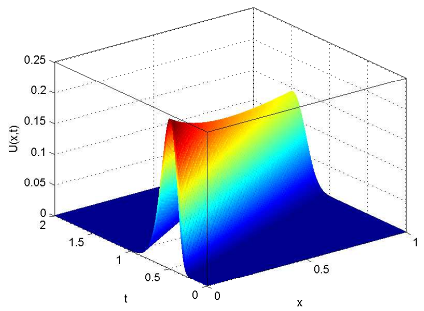

Figure 10: Exact solution of (3.3) using $\epsilon = 1, \gamma = 0.01, x_0 = -0.5, \sigma_0 = 0.025, h = 0.01, \tau = 0.001, T = 2$ *and* $L = 1$

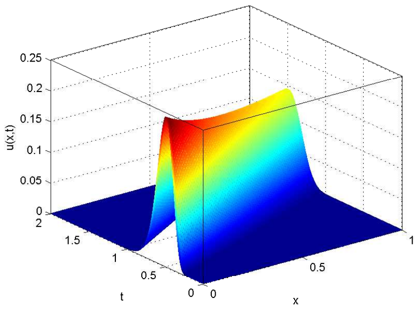

Figure 11: Numerical solution of (3.3) using $\epsilon = 1, \gamma = 0.01, x_0 = -0.5, \sigma_0 = 0.025, h = 0.01, \tau = 0.001, T = 2$ *and* $L = 1$

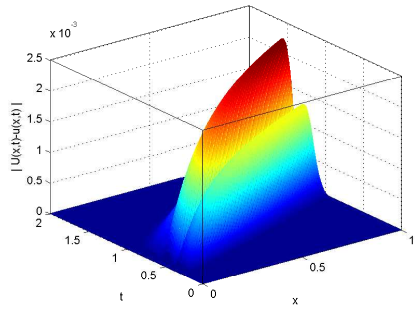

Figure 12: Absolute error of (3.3) using $\epsilon = 1, \gamma = 0.01, x_0 = -0.5, \sigma_0 = 0.025, h = 0.01, \tau = 0.001, T = 2$ *and* $L = 1$

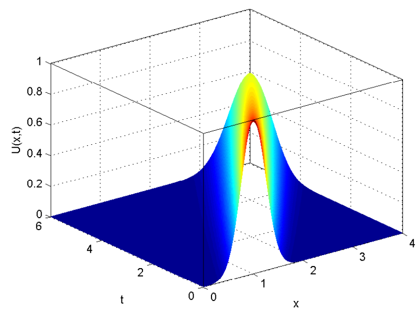

Figure 13: Exact solution of (3.3) using $\epsilon = 0.5, \gamma = 0.01, x_0 = 1, \sigma_0 = 0.25, h = 0.01, \tau = 0.001, T = 6$ *and* $L = 4$

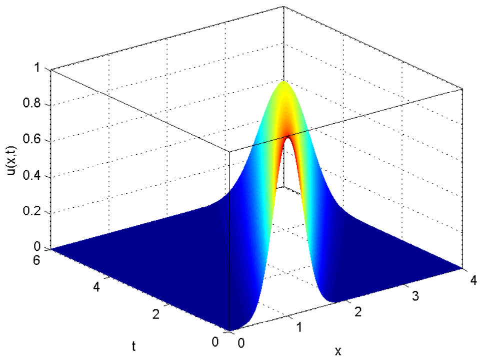

Figure 14: Numerical solution of (3.3) using $\epsilon = 0.5, \gamma = 0.01, x_0 = 1, \sigma_0 = 0.25, h = 0.01, \tau = 0.001, T = 6$ *and* $L = 4$

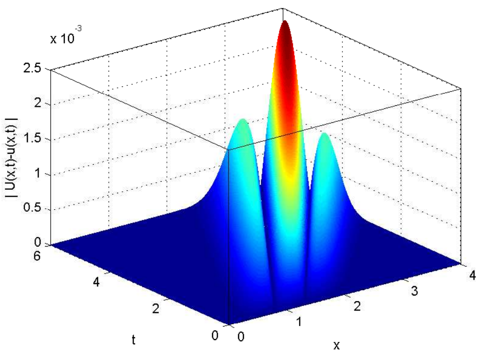

Figure 15: Absolute error of (3.3) using $\epsilon = 0.5, \gamma = 0.01, x_0 = 1, , \sigma_0 = 0.25, h = 0.01, \tau = 0.001, T = 6$ *and* $L = 4$

Table 4: Errors of (3.3) using $\epsilon = 1, \gamma = 0.01, x_0 = -0.5, \sigma_0 = 0.025, h = 0.01, \tau = 0.001, T = 2$ *and* $L = 1$

	$t = 0.5$	$t = 1$	$t = 1.5$	$t = 2$
$x = 0.1$	1.8144e-04	4.7594e-05	9.3582e-09	2.3923e-13
$x = 0.2$	9.2820e-04	2.8199e-04	2.3797e-07	1.1222e-11
$x = 0.3$	4.4890e-04	4.5771e-04	3.0825e-06	2.9660e-10
$x = 0.4$	4.7669e-05	5.0619e-04	2.4601e-05	5.4092e-09
$x = 0.5$	1.4460e-06	2.0176e-03	1.2517e-04	7.1842e-08
$x = 0.6$	1.3372e-08	1.2396e-03	3.9731e-04	7.0828e-07
$x = 0.7$	3.8782e-11	8.9995e-04	7.0882e-04	5.2113e-06
$x = 0.8$	3.7521e-14	1.3376e-03	3.8544e-04	2.8536e-05
$x = 0.9$	1.3477e-17	5.9735e-04	1.0301e-03	1.1504e-04
L_2	3.2786e-05	9.5514e-05	6.4029e-05	6.8806e-06
L_∞	9.4414e-04	2.1034e-03	2.0682e-03	2.5806e-04

Table 5: Errors of (3.3) using $\epsilon = 0.5, \gamma = 0.01, x_0 = 1, \sigma_0 = 0.25, h = 0.01, \tau = 0.001, T = 6$ *and* $L = 4$

	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$	$t = 6$
$x = 0.5$	2.1148e-05	3.3534e-07	2.2440e-09	1.0190e-11	3.7126e-14	1.1739e-16
$x = 1$	5.5666e-04	8.7226e-05	2.2966e-06	2.6200e-08	1.8511e-10	9.6722e-13
$x = 1.5$	1.2499e-03	8.4289e-04	1.9575e-04	8.1979e-06	1.4647e-07	1.5421e-09
$x = 2$	5.6004e-04	1.8280e-03	9.2924e-04	3.2799e-04	2.0487e-05	5.3775e-07
$x = 2.5$	5.5020e-05	6.9113e-04	2.1162e-03	8.9925e-04	4.6445e-04	4.0761e-05
$x = 3$	9.1453e-08	1.8141e-04	6.2597e-04	2.2623e-03	8.0872e-04	5.9138e-04
$x = 3.5$	5.2812e-12	1.2693e-06	3.4891e-04	4.8353e-04	2.3326e-03	6.9051e-04
L_2	8.0990e-05	1.2384e-04	1.4893e-04	1.6362e-04	1.6128e-04	1.1192e-04
L_∞	1.3097e-03	1.8976e-03	2.1831e-03	2.3235e-03	2.3878e-03	2.1097e-03

Example 3.4 ([15])**.** As final test problem, (1.1) has exact solution

$$u(x,t) = \frac{1}{\sqrt{s}} e^{\frac{-50(x-t)^2}{s}}, \quad s = 1 + 200\gamma t$$

with $\epsilon = 1, \gamma = 1, h = 0.02, \tau = 0.0001, T = 5, L = 1$ and the following initial condition

$$\phi(x) = \frac{1}{\sqrt{s}} e^{\frac{-50x^2}{s}}, \quad s(diffusion\ number) = 1.$$

The boundary conditions can be obtained from the exact solution. The exact solution is shown in Figure 16 and the approximated is shown in Figure 17. Absolute error is shown in Figure 18. The absolute errors for some values of t and error norms are reported in Table 6.

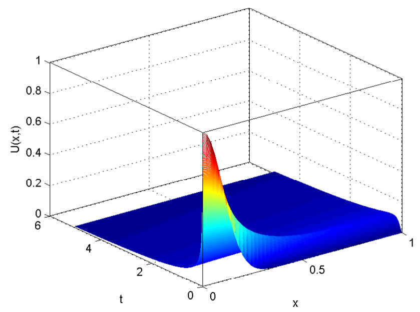

Figure 16: Exact solution of (3.4) using $\epsilon = 1, \gamma = 1, h = 0.02, \tau = 0.0001, T = 5, L = 1$

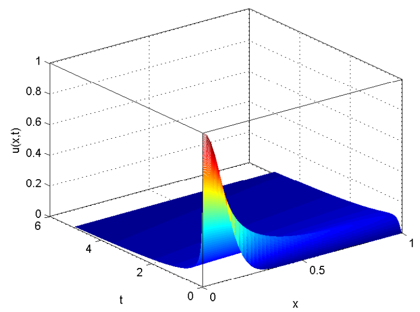

Figure 17: Numerical solution of (3.4) using $\epsilon = 1, \gamma = 1, h = 0.02, \tau = 0.0001, T = 5, L = 1$

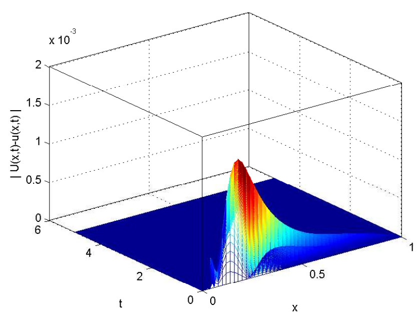

Figure 18: Absolute error of (3.4) using $\epsilon = 1, \gamma = 1, h = 0.02, \tau = 0.0001, T = 5, L = 1$

Table 6: Errors of (3.4) using $\epsilon = 1, \gamma = 1, h = 0.02, \tau = 0.0001, T = 5$ *and* $L = 1$

	$t = 0.1$	$t = 1$	$t = 2$	$t = 5$
$x = 0.1$	4.4353e-05	1.4759e-07	1.5244e-05	1.8722e-09
$x = 0.5$	2.6284e-04	6.1066e-07	8.8292e-08	7.7043e-09
$x = 0.9$	1.0684e-04	2.7987e-07	4.4067e-08	1.6307e-05
L_2	2.5970e-05	6.3033e-08	9.2375e-09	8.1088e-10
L_∞	2.6750e-04	6.1978e-07	9.0535e-08	7.9055e-09

4 Conclusions

In this study, the cubic B-spline quasi-interpolation (BSQI) method is used for solving convection-diffusion equation. The numerical solutions of some test problems are compared with the exact solutions to convince us that the proposed scheme of numerical approximation seems to be accurate and dependable. The implementation of the present method is a very easy, acceptable, and valid scheme.

References

[1] M. Dehghan, Weighted finite difference techniques for the one-dimensional advection-diffusion equation, Appl. Math. Comput. 147 (2004) 307-319.

[2] M. Sari, G. Graslan, A. Zeytinoglu, High-Order finite difference schemes for solving the advection-diffusion equation, Math. Comput. Appl. 15 (3) (2010) 449-460.

[3] A. Mohebbi, M. Dehghan, High-order compact solution of the one-dimensional heat and advection-diffusion equations, Appl. Math. Model. 34 (2010) 3071-3084.

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

[5] H. Karahan, Implicit finite difference techniques for the advection-diffusion equation using spreadsheets, Adv. Eng. Software 37 (2006) 601-608.

[6] H.N.A. Ismail, E.M.E. Elbarbary, G.S.E. Salem, Restrictive Taylor's approximation for solving convection-diffusion equation, Appl. Math. Comput. 147 (2004) 355-363.

[7] M.K. Kadalbajoo, L.P. Tripathi, A. Kumar, A cubic B-spline collocation method for a numerical solution of the generalized Black-Scholes equation, Mathematical and Computer Modelling 55 (2012) 1483-1505.

[8] J. Goh, A.A. Majid, A.I.M. Ismail, A quartic B-spline for second-order singular boundary value problems, Comp. Math. Appl. 64 (2012) 115-120.

[9] S.A. Khuri, A. Sayfy, A spline collocation approach for a generalized parabolic problem subject to non-classical conditions, Appl. Math. Comput. 218 (2012) 9187-9196.

[10] R.C. Mittal, R.K. Jain, Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method, Appl. Math. Comput. 218 (2012) 7839-7855.

[11] H.M. El-Hawary, S.M. Mahmoud, Spline collocation methods for solving delay-differential equations, Appl. Math. Comput. 146 (2003) 359-372.

[12] C.G. Zhu, R.H. Wang, Numerical solution of Burgers' equation by cubic B-spline quasi-interpolation, Appl. Math. Comput. 208 (2009) 260-272.

[13] M. Dosti, A. Nazemi, Solving one-dimensional hyperbolic telegraph equation using cubic B-spline quasi-interpolation, International Journal of Mathematical & Computer Sciences 7 (2011) Issue 2, p57.

[14] R.G. Yuab, R.H. Wanga, C.G. Zhu, A numerical method for solving KdV equation with multilevel B-spline quasi-interpolation, Applicable Analysis: An International Journal 92 (8) (2012) 1682-1690.

[15] R.C. Mittal, R.K. Jain, Redefined cubic B-splines collocation method for solving convection-diffusion equations, Appl. Math. Model. 36 (2012) 5555-5573.

[16] M.K. Kadalbajoo, P. Arora, Taylor-Galerkin B-spline finite element method for the one dimensional advection-diffusion equation, Numer. Methods Partial Diff. Eqs. 26 (5) (2009) 2006-1223.

[17] P. Sablonniere, Univariate spline quasi-interpolants and applications to numerical analysis, Rendiconti del Seminario Matematico 63 (2005) 211-222.

(Received 11 September 2013)
(Accepted 13 March 2014)