A comparative study of numerical schemes for convection-diffusion equation

V.S. Aswin\textsuperscript{a,}\textsuperscript{*}, Ashish Awasthi\textsuperscript{a}, C. Anu\textsuperscript{b}

\textsuperscript{a}National Institute of Technology, Department of Mathematics, Calicut, India
\textsuperscript{b}Zamorin’s Guruvayurappan College, Department of Mathematics, Calicut, India

Abstract

In this paper, three different numerical schemes are described to approximate the solution of the convection-diffusion equation. The methods are based on differential quadrature and finite difference. In the first scheme, time derivative is approximated using forward difference and the space derivatives using polynomial based differential quadrature method (PDQM). In the second scheme, the discretization of the time and space derivatives are done using PDQM and central difference respectively, while in the third scheme only PDQM is used for the discretization of both time and space derivatives. The validation and comparison of the schemes are done through the simulation of two classic examples of convection-diffusion problem having known exact solution. It is found that the numerical schemes are in excellent agreement with the exact solution. We conclude with the realization that the third scheme, i.e. PDQM in time and space produce more accurate results among these three schemes.

\textsuperscript{c}\textregistered\textsuperscript{2}2015 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Keywords: convection-diffusion equation; polynomial based differential quadrature method; finite difference method; Chebyshev-Gaus-Lobatto grid

1. Introduction

The exchange of heat, mass and momentum are considered to be the fundamental transfer phenomena in the universe. Hence it has great importance in various engineering disciplines and applied science. The mathematical framework for heat and mass transfer are of same kind, basically encompass by advection and diffusion effects. Such general scalar transport equations are broadly termed as convection-diffusion equation.

Consider one dimensional time dependent convection-diffusion equation with constant coefficients for a general scalar variable $\varphi$ subjected to appropriate initial and boundary conditions.

$$\frac{\partial \varphi}{\partial t} + a \frac{\partial \varphi}{\partial x} - D \frac{\partial^2 \varphi}{\partial x^2} = 0, \quad 0 < x < L, \quad 0 < t \leq T$$ (1)

$$\varphi(x, 0) = f(x), \quad 0 \leq x \leq L$$ (2)

\textsuperscript{*} Corresponding author.

E-mail address: aswin.p140044ma@nitc.ac.in(V.S. Aswin), aawasthi@nitc.ac.in(A. Awasthi), anuccheruvalath@gmail.com(C. Anu)
\( \varphi(0, t) = g_1(t), \varphi(L, t) = g_2(t), \quad 0 \leq t \leq T \) (3)

In this prototype equation the parameters \( a \) and \( D \) stand for phase speed and the diffusion coefficient respectively, and both are assumed to be positive. This is one of the basic linear partial differential equations, where fluid flow is important. Convection-dominated diffusion problems have more applications. Kaya [1] has listed a few of the applications and developed a polynomial based differential quadrature method. He has compared the results with implicit and explicit finite difference methods and found that DQM gives better results. Appadu et al. [2] have conducted a computational based study of three numerical schemes based on the finite difference method containing third and fourth order upwind scheme and non-standard finite difference scheme. A finite difference scheme is also analysed by Salkuyeh [3]. Analytical and finite element solution of Eq. (1) with \( f(x) = -\sin(\pi x) \) and homogeneous boundary conditions was given in Mojtabi and Deville [4]. The methods like upwind finite volume approximation [5], cubic B-splines collection method [6], Haar wavelet [7], restrictive Taylors approximation [8] and operator splitting algorithms [9] are also employed for convection-diffusion problems.

In this study, Eq. (1) is solved using PDQM. This method was proposed by Bellman and Casti [10]. Quan and Chang [11] and Shu [12] have introduced new and better formulas for determining weighting coefficients. In section 2, formulations of the numerical schemes are described with the help of Eq. (4) for the approximation of derivatives in DQM. Numerical experiments and discussion are in section 3 followed by conclusions, summarized in section 4.

\[
\frac{\partial \varphi}{\partial t} \bigg|_{x_i} = \sum_{j=1}^{N} A_{ij}^{(r)} \varphi(x_i), \quad i = 1, 2, \ldots, N
\] (4)

2. Numerical schemes

This section presents the formulation of three numerical schemes for convection-diffusion equation using a combination of finite difference and polynomial based differential quadrature methods. Finite difference method employed on a uniform computational grid while Chebyshev-Gaus-Lobatto (CGL) grid points are selected for differential quadrature implementation. The formula for the calculation of CGL grid points is given in Kaya [1] as

\[
x_i = \frac{r_i - r_1}{r_R - r_1}
\]

where

\[
r_i = \frac{1}{2} \left(1 - \cos \frac{i - 1}{R - 1} \pi \right)
\]

\( R \) denotes the number of grid points; \( N \) in the time domain and \( M \) in the space domain.

In the present study, weighting coefficients in PDQM are computed using Shus general approach [12,13]. According to Shus approach, weighting coefficients of first and second order derivatives can compute using the formulas

\[
A_{ij}^{(1)} = \frac{L_1^{(1)}(x_i)}{x_i - x_j} \frac{L_1^{(1)}(x_j)}{x_i - x_j}
\]

\[
A_{ii}^{(1)} = -\sum_{j=1}^{N} A_{ij}^{(1)}, \quad i \neq j
\]

\[
A_{ij}^{(2)} = 2A_{ij}^{(1)} \left( A_{ii}^{(1)} - \frac{1}{x_i - x_j} \right)
\]

\[
A_{ii}^{(2)} = -\sum_{j=1}^{N} A_{ij}^{(2)}, \quad i \neq j
\] (7) (8)

Where \( L_1^{(1)}(x_i) = \prod_{k=1}^{N} (x_i - x_k), \quad i \neq k \) is a first derivative of polynomial function of degree \( N \).

2.1. Scheme 1: Finite Difference in Time and Differential Quadrature in Space (FDTDQS)

The first step in the formulation of FDTDQS scheme consists of discretization of time derivative of Eq. (1) using weighted finite difference formula. Eq. (1) becomes

\[
\frac{u^{n+1} - u^n}{\Delta t} = (1 - \theta)[D u_{xx}^n + a u_x^t] + \theta[D u_{xx}^{n+1} + a u_x^{n+1}]
\] (9)
where \( u^n(x) \approx \varphi(x, t_n) \) and \( 0 \leq \theta \leq 1 \) is a parameter and \( n = 1, 2, \ldots, N - 1 \). Now apply differential quadrature method for the discretization of space derivatives in Eq. (9)

\[
U^{n+1}_m + \theta \Delta t \left( a \sum_{i=1}^{M} A_{mi} U^{n+1}_i - D \sum_{i=1}^{M} A_{mi} U^{n+1}_i \right) = U^n_m - (1 - \theta) \Delta t \left( a \sum_{i=1}^{M} A_{mi} U^n_i - D \sum_{i=1}^{M} A_{mi} U^n_i \right)
\]

(10)

\( m = 1, 2, \ldots, M, U^n_m \approx u^n(x_m) \); with initial and boundary conditions

\[
U^1_m = f(x_m) = f(m), \quad U^n_1 = g_1(t_n) = g_1(n), \quad U^n_M = g_2(t_n) = g_2(n)
\]

(11)

After rearranging and applying boundary conditions, Eq. (10) yields to the FDTDQS scheme;

\[
U^{n+1}_m + \theta \sum_{i=2}^{M-1} A_{mi} U^{n+1}_i = K^n_m
\]

(12)

where \( A_{mi} = \Delta t \left( a A_{mi} - DA_{mi} \right) \) and \( K^n_m = U^n_m - (1 - \theta) \sum_{i=1}^{M} A_{mi} U^n_i - \theta (A_{m1} g_{1(n+1)} + A_{mM} g_{2(n+1)}) \)

2.2. Scheme 2: Differential Quadrature in Time and Finite Difference in Space (DQTFDS)

Consider Eq. (1) and the use of central difference formula for the discretization of space derivatives forms system of ordinary differential equation in time.

\[
\frac{\partial u_m}{\partial t} = -a \frac{u_{m+1} - u_{m-1}}{2\Delta x} + D \frac{u_{m+1} - 2u_m + u_{m-1}}{\Delta x^2}
\]

(13)

for \( m = 2, 3, \ldots, M - 1 \) and \( u_m(t) \approx \varphi(x_m, t) \). These system of differential equations are solved using differential quadrature method, and it yields to the DQTFDS scheme as

\[
\sum_{j=2}^{N} B_{nj}^{(1)} U^n_m = (q - p) U^n_{m+1} - 2q U^n_m + (q + p) U^n_{m-1} - B_{n1}^{(1)} f(m)
\]

(14)

where \( p = a/2\Delta x \) and \( q = D/\Delta x^2 \) and \( U^n_m \approx u_m(t_n) \) for \( n = 2, 3, \ldots, N \) with initial and boundary conditions given in Eq. (11).

2.3. Scheme 3: Differential Quadrature in Time and Differential Quadrature in Space (DQTDQS)

Discretizing time and space derivatives of Eq. (1) using differential quadrature method yields

\[
\sum_{j=2}^{N} B_{nj}^{(1)} U^n_m + a \sum_{i=1}^{M} A_{mi} U^n_i - D \sum_{i=1}^{M} A_{mi} U^n_i = 0
\]

(15)

Where \( U^n_m \approx \varphi(x_m, t_n) \). The discretization of initial and boundary conditions are found to be similar to that of FDTDQS Eq. (11). In order to generate computational model of DQTDQS, determine and apply Eq. (11), leads to

\[
\sum_{j=2}^{N} B_{nj}^{(1)} U^n_m + \sum_{i=2}^{M-1} A_{mi} U^{n+1}_i = K^n_m
\]

(16)

where \( A_{mi} = \left( a A_{mi} - DA_{mi} \right) \) and \( K^n_m = \left( B_{n1}^{(1)} f(m) + A_{m1} g_{1(n)} + A_{mM} g_{2(n)} \right) \) for \( m = 2, 3, \ldots, M - 1 \) and \( n = 2, 3, \ldots, N \).
3. Numerical experiments and discussions

In this section, illustration of the accuracy of proposed schemes has conducted through estimating error in $L_2$ and $L_\infty$ norms, provided in Eq. (17) and (18) for specific time level.

\[
L_2 = \left( \frac{1}{M} \sum_{j=1}^{M} |\phi_j^{(\text{exact})} - U_j^{(\text{num})}|^2 \right)^{1/2}
\]

\[
L_\infty = \|\phi^{(\text{exact})} - U^{(\text{Num})}\|_\infty = \max_j |\phi_j^{(\text{exact})} - U_j^{(\text{num})}|
\]

The computation of numerical rate of convergence of the schemes have carried out using the formula

\[
\text{ROC} \approx \frac{\log(E(N_2)/E(N_1))}{\log(N_1/N_2)}
\]

where $E(N_j)$ is $L_\infty$ with $N_j$ grid.

Major aspects of this section have originated with implementation of schemes in MATLAB. In order to demonstrate the efficiency and the analogy of the schemes, we opt and solved two well-posed convection-diffusion problems.

3.1. Problem 1

Consider the convection-diffusion equation Eq. (1) with initial condition

\[
\phi(x, 0) = \sin(2\pi x), \quad 0 \leq x \leq 1
\]

and boundary conditions

\[
\phi(0, t) = \exp(-D4\pi^2t) \sin(-2\pi at), \quad \phi(1, t) = \exp(-D4\pi^2t) \sin(2\pi(1 - at)), \quad 0 \leq t \leq T
\]

Table 1. Error in $L_2$ and $L_\infty$ norms at $T = 0.5$ with $M = N = 20$ for Pechlet number $Pe = a/D = 20$.

<table>
<thead>
<tr>
<th>D</th>
<th>FDTDQS $L_2$</th>
<th>FDTDQS $L_\infty$</th>
<th>DQTDQS $L_2$</th>
<th>DQTDQS $L_\infty$</th>
<th>DQTDQS $L_2$</th>
<th>DQTDQS $L_\infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>2.16E-03</td>
<td>4.58E-03</td>
<td>6.21E-03</td>
<td>1.41E-02</td>
<td>6.61E-07</td>
<td>1.96E-06</td>
</tr>
<tr>
<td>0.01</td>
<td>3.23E-05</td>
<td>5.39E-05</td>
<td>5.73E-03</td>
<td>9.46E-03</td>
<td>7.94E-09</td>
<td>2.64E-08</td>
</tr>
<tr>
<td>0.001</td>
<td>3.97E-08</td>
<td>6.46E-08</td>
<td>7.25E-04</td>
<td>1.12E-03</td>
<td>1.94E-13</td>
<td>7.59E-13</td>
</tr>
<tr>
<td>0.0001</td>
<td>4.22E-11</td>
<td>6.54E-11</td>
<td>7.49E-05</td>
<td>1.15E-04</td>
<td>2.32E-15</td>
<td>6.74E-15</td>
</tr>
</tbody>
</table>

Exact solution of the problem is given as

\[
\phi(x, t) = \exp(-D4\pi^2t) \sin(2\pi(x - at))
\]

The simulation outcomes of this problem can be found in Table 2 and in Fig. 1 for a diffusion coefficient $D = 0.05$ and convection velocity $a = 1$. We found FDTDQS gives better result with $\theta = 0.5$ and hence used for all the computations. Fig. 1 contains two graphs; first one illustrates the physical behaviour of problem 1 generated using DQTDQS scheme on CGL mesh. While the second graph provide visual idea about the accuracy of the schemes. It is noted that the exact and DQTDQS scheme results are indistinguishable even in the popup plot. CGL grid is modified in order to analyse the numerical results at various mesh points in the schemes and is given in Table 2. $-\log(L_2)$ versus $t$ has plotted in first graph of Fig. 3, which is helpful to understand the behaviour of error in each time steps. Error in $L_2$ and $L_\infty$ are computed at $T = 0.5$ for various values of $a$ and $D$; and showcased it in Table 1. Max error and rate of convergence of three schemes are included in Table 3, together with $L_2$ error for $D = 0.05$ and $a = 1.0$. All these analysis point out that the scheme DQTDQS gives highly accurate results and then FDTDQS for problem 1.
Fig. 1. Physical behaviour of Numerical solution using DQTDQS (left), numerical and exact solutions at T=0.5 (right) using M&N = 20.

Table 2. Comparison of exact and numerical solutions at various mesh points using M&N = 20.

<table>
<thead>
<tr>
<th>x</th>
<th>t</th>
<th>Problem 1</th>
<th>DQTDQS</th>
<th>DQTFDS</th>
<th>DQTDQS</th>
<th>Exact</th>
<th>Problem 2</th>
<th>DQTDQS</th>
<th>DQTFDS</th>
<th>DQTDQS</th>
<th>Exact</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>0.4</td>
<td>-0.3662</td>
<td>-0.3787</td>
<td>-0.3673</td>
<td>-0.3673</td>
<td>0.0011</td>
<td>0.0014</td>
<td>2.6e-04</td>
<td>2.2e-04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td></td>
<td>-0.2509</td>
<td>-0.2349</td>
<td>-0.2475</td>
<td>-0.2475</td>
<td>0.0859</td>
<td>0.0993</td>
<td>0.0912</td>
<td>0.0913</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td></td>
<td>0.0616</td>
<td>0.0742</td>
<td>0.0637</td>
<td>0.0637</td>
<td>0.1855</td>
<td>0.1778</td>
<td>0.1798</td>
<td>0.1798</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td>0.1395</td>
<td>0.1376</td>
<td>0.1389</td>
<td>0.1389</td>
<td>0.0379</td>
<td>0.0339</td>
<td>0.0383</td>
<td>0.0383</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.50</td>
<td>0.4</td>
<td>0.2775</td>
<td>0.3333</td>
<td>0.2669</td>
<td>0.2669</td>
<td>1.9e-06</td>
<td>2.0e-05</td>
<td>3.3e-05</td>
<td>2.3e-10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td></td>
<td>-0.1759</td>
<td>-0.1255</td>
<td>-0.1798</td>
<td>-0.1798</td>
<td>0.0016</td>
<td>0.0012</td>
<td>4.1e-04</td>
<td>3.9e-04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td></td>
<td>-0.1995</td>
<td>-0.2003</td>
<td>-0.1961</td>
<td>-0.1961</td>
<td>0.0559</td>
<td>0.0457</td>
<td>0.0582</td>
<td>0.0582</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td>-0.0034</td>
<td>-0.0255</td>
<td>0.0000</td>
<td>-0.0000</td>
<td>0.1737</td>
<td>0.1596</td>
<td>0.1741</td>
<td>0.1741</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.75</td>
<td>0.4</td>
<td>0.3702</td>
<td>0.3477</td>
<td>0.3673</td>
<td>0.3673</td>
<td>1.8e-08</td>
<td>6.7e-07</td>
<td>3.3e-07</td>
<td>1.7e-19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td></td>
<td>0.2590</td>
<td>0.2745</td>
<td>0.2475</td>
<td>0.2475</td>
<td>8.3e-06</td>
<td>1.9e-05</td>
<td>1.9e-05</td>
<td>1.2e-08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td></td>
<td>-0.0573</td>
<td>-0.0316</td>
<td>-0.0637</td>
<td>-0.0637</td>
<td>0.0016</td>
<td>0.0016</td>
<td>4.6e-04</td>
<td>4.3e-04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td>-0.1408</td>
<td>-0.1326</td>
<td>-0.1389</td>
<td>-0.1389</td>
<td>0.0383</td>
<td>0.0361</td>
<td>0.0382</td>
<td>0.0383</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.2. Problem 2

The initial and boundary conditions of the convection-diffusion equation Eq. (1) over the domain [0, 1] with \(a = 1.0\) and \(D = 0.01\), are considered as in [14,15].

\[
\phi(x,0) = \exp \left( -\frac{(x+0.5)^2}{0.00125} \right) \tag{23}
\]

\[
\phi(0,t) = \frac{0.025}{\sqrt{0.000625 + 0.02t}} \exp \left( -\frac{(0.5-t)^2}{0.000125 + 0.04t} \right) \tag{24}
\]

\[
\phi(1,t) = \frac{0.025}{\sqrt{0.000625 + 0.02t}} \exp \left( -\frac{(1.5-t)^2}{0.000125 + 0.04t} \right) \tag{25}
\]

The analytical solution of this problem is as follows

\[
\phi(x,t) = \frac{0.025}{\sqrt{0.000625 + 0.02t}} \exp \left( -\frac{(x+0.5-t)^2}{0.000125 + 0.04t} \right) \tag{26}
\]

The comparisons of the numerical and exact solutions are presented in Table 2 and in the second graph in Fig. 2. The result of DQTDQS seems to be closer to the exact solution. The insights from the numerical error analysis
presented in Table 3 and in the second graph of Fig. 3 are also agreed with our observation. First graph in Fig. 2 exhibits the solution of problem 2 using DQTDQS scheme in a square domain $0 \leq x, t \leq 1$. 

![Graph](image1)

![Graph](image2)

**Fig. 2.** Physical behaviour of Numerical solution using DQTDQS (left), numerical and exact solutions at $T = 0.5$ (right) using $M\&N = 20$.

**Table 3.** Rate of convergence and error in $L^2$ and $L^\infty$ norms at $T = 0.5$ with $M = N$

<table>
<thead>
<tr>
<th>Prob</th>
<th>N</th>
<th>$L^2$</th>
<th>$L^\infty$</th>
<th>ROC</th>
<th>$L^2$</th>
<th>$L^\infty$</th>
<th>ROC</th>
<th>$L^2$</th>
<th>$L^\infty$</th>
<th>ROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>0.0048</td>
<td>0.0113</td>
<td>...</td>
<td>2.011</td>
<td>0.0189</td>
<td>0.0319</td>
<td>5.21e-4</td>
<td>0.0014</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.0022</td>
<td>0.0050</td>
<td>...</td>
<td>2.016</td>
<td>0.0092</td>
<td>0.0165</td>
<td>2.292</td>
<td>1.92e-7</td>
<td>5.78e-7</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.0012</td>
<td>0.0028</td>
<td>...</td>
<td>2.016</td>
<td>0.0092</td>
<td>0.0165</td>
<td>2.292</td>
<td>1.92e-7</td>
<td>5.78e-7</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>0.0023</td>
<td>0.0071</td>
<td>...</td>
<td>0.0072</td>
<td>0.0170</td>
<td>0.0170</td>
<td>7.33e-4</td>
<td>0.0018</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>9.06e-4</td>
<td>0.0028</td>
<td>2.2948</td>
<td>0.0047</td>
<td>0.0126</td>
<td>0.0126</td>
<td>0.7387</td>
<td>6.10e-5</td>
<td>1.07e-4</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>5.18e-4</td>
<td>0.0015</td>
<td>2.1696</td>
<td>0.0030</td>
<td>0.0072</td>
<td>0.0072</td>
<td>1.9453</td>
<td>1.23e-6</td>
<td>4.28e-6</td>
</tr>
</tbody>
</table>

![Graph](image3)

**Fig. 3.** Error exploration of the schemes for problem 1 (left) and problem 2 (right).
4. Conclusion

In this paper, we considered three numerical schemes for convection-diffusion equation based on combinations of polynomial based differential quadrature and finite difference methods. Among the three schemes FDTDQS, DQTFDS and DQTDQS, DQTDQS scheme produced most accurate results in compare with other two. FDTDQS produced better results in compare to DQTFDS. One interesting fact explored from the tables and figures of two examples is that all three schemes giving good results and the accuracy of the results improved as more number of derivatives are approximated using the differential quadrature method.

Large number of nodes leads to an ill-conditioned system [12] in PQDM. One can achieve higher resolution by combining the results of multi-simulations with different nodes, with same accuracy and speed. These methods can be extended to any linear and linearized nonlinear higher dimensional problems.

Acknowledgements

The authors are very grateful to Kerala State Council for Science, Technology and Environment (KSCSTE) for their financial support. We are thankful for the valuable comments and suggestions of the reviewers.

References