Extensions to Higher-Dimensions of an Unconditionally Stable ADE Scheme for the Convection-Diffusion Equation

Martin Guay, Fabrice Colin, Richard Egli*

21/09/2010

Département d'informatique
Université de Sherbrooke
lemailamartin@gmail.com
richard.egli@usherbrooke.ca

Département de mathématiques et d'informatique
Université de Laurentienne
fcolin@cs.laurentian.ca

- Technical Report No. 31 -

Abstract

An unconditionally stable alternating direction explicit scheme (ADE) to solve the one-dimensional unsteady convection-diffusion equation was developed by J. Xie, Z. Lin and J. Zhou in [6]. Aside from being explicit and unconditionally stable, the method is straightforward to implement. In this paper we show extensions of this scheme to higher-dimensions of the convection-diffusion equation subject to Dirichlet boundary conditions. By expressing the equation with a local series expansion over a rectangular grid, a linear system of symbolic equations is obtained which is tedious to solve for manually and we addressed this challenge using symbolic computation. The solutions obtained are explicit closed-form formulas which are then used to iteratively solve the unsteady convection-diffusion equation by traversing the discrete grid in an alternating direction fashion. Finally, extensions to higher dimensions can be easily deduced from the 2D formulas. We conclude the paper with numerical simulation results for diffusion and convection-diffusion problems compared to analytical solutions showing the performance of the method and its numerical stability.

Keywords: Convection-Diffusion Equations, Alternating Direction Explicit(ADE), Unconditionally Stable.

*Corresponding author: Martin Guay; lemailamartin@gmail.com

Contents

List of Figures

1 Introduction

The convection-diffusion equation is present in many fields of applications and numerical schemes to solve it are abundant. The choice of a particular method is usually motivated by its features and strongly influenced by the nature and constraints of the application. Numerical methods to solve the convection-diffusion equation which possess both the advantages of explicit and implicit algorithms are rare. Gourlay[2], Abdulah[1] and Zhang[7] all proposed explicit and unconditionally stable finite difference methods based on Saul'yev scheme[4]. We show in this paper extensions to a method which has nothing to do with finite differences yet remains simple to implement. Others proposed explicit methods with the desired ease of implementation but are limited to solving only the transport equation as in [5, 3], the diffusion equation as in [4, 3] and not the convection-diffusion equation as a whole. On the other hand, we also consider in this paper the importance of discussing the difficulties encountered when considering extending the method to higher-orders of accuracy than $O(\Delta x^2, \Delta t)$[6]. As to clearly elucidate the characteristics of the proposed method, we summarize its features in the following lists.

In short, the method illustrated in this paper is :

- straightforward to implement,
- unconditionally stable,
- explicit and therefore requires less computational time and storage.

Unfortunately, it is limited to

- solve the convection-diffusion equation,
- achieve at most a theoretical second order accuracy in space and first order accuracy in time.

The rest of the paper consists of a derivation of the proposed method for the two-dimensional unsteady convection-diffusion equation with illustrations of experimental simulation results compared to analytical solutions. In order for the proposed method to be easily tested or compared, the Mathematica notebook was made available on this website[1]. For consistency purposes we employ the same notation as in [6] where the convection-diffusion equation refers to

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \left(\frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y} \right) = D \left(\frac{\partial \phi^2}{\partial x^2} + \frac{\partial \phi^2}{\partial y^2} \right) \text{ subject to } \phi(\mathbf{x}, t) = f(t) \text{ in } \partial\Omega \tag{1.1}$$

where $\phi = \phi(\mathbf{x}, t) : \Omega \times [0, T] \rightarrow \mathbb{R}$ and $\mathbf{u} = \mathbf{u}(\mathbf{x}, t) : \Omega \times [0, T] \rightarrow \mathbb{R}^2$ are both functions of space $\mathbf{x}$ and time t and D is a constant diffusive coefficient. f is a given piecewise function on the 4 edges of Ω which forms $\partial\Omega$. The domain $\Omega : (0, 1) \times (0, 1)$ is divided as usual into $X \times Y$ cells of spacing $\mathbf{h} = (\frac{1}{X}, \frac{1}{Y})$ for x and y directions respectively. The solution at every grid point is denoted $\phi^k_{i,j}$ where $i \in 1, 2, \cdots, X$

[1](Deprecated, send e-mail to first author)`http://pages.usherbrooke.ca/mguay/ADE`

and $j \in 1, 2, \cdots, Y$. $\phi^k_{i,j}$ can be approximated in the spatial subdomain $[x_{i-1}, x_{i+1}] \times [y_{j-1}, y_{j+1}]$, by the two dimensional nth-order Taylor series expansion:

$$\phi(\chi, \gamma, \tau) = \sum_{i=0,j=0}^{N} a_{i,j}(\tau)\chi^i\gamma^j \tag{1.2}$$

where $\chi = x - x_i, \gamma = y - y_j, \tau = t - t_k$ and $a_{i,j}(\tau)$ are the unknown coefficients forming a function of space and time in the subdomain $[x_{i-1}, x_{i+1}] \times [y_{j-1}, y_{j+1}]$. By expressing equation (1.1) with the 2nd-order Taylor series approximation and analytically integrating the $\dot{a}_{i,j}$ terms with temporal derivatives, an approximation of $\phi(\chi, \gamma, \tau)$ satisfying the equation is derived. The remaining unknown $a_{i,j}$ coefficients are found by judiciously choosing elements from the neighborhood of $\phi^k_{i,j}$ and building a system of linear equations. This system is solved symbolically in order to find the mapping of the $a_{i,j}$ coefficients to local $\phi_{i,j}$ samples on the grid.

2 Derivation

The main challenge in extending the method to higher dimensions resides in the algebraic linear system which is required to be solved. Solving this system by hand, even for the two-dimensional case, is time consuming and prone to errors. To address this issue, we used symbolic computation through *Mathematica* and as mentioned earlier, made the code to obtain the two-dimensional formulas available on this web page (Deprecated, send e-mail to first author)`http://pages.usherbrooke.ca/mguay/ADE`.

Although the derivation process can become complex when performed by hand, the resulting method for the convection-diffusion equation is consists of two explicit formulas for the approximation of the equation's solution which can be used to solve the equation by traversing the computational grid in two alternate directions: left-to-right and right-to-left.

2.1 Local series approximation of the convection-diffusion equation

The first step of the local series expansion method consists in substituting the Taylor series expansion (1.2) into the convection-diffusion equation (1.1) which leads to:

$$\sum_{i=0}^{N}\sum_{j=0}^{N} \dot{a}_{i,j}(\tau)\chi^i\gamma^j + \mathbf{u} \cdot \left(\sum_{i=0}^{N-1}\sum_{j=0}^{N}(i+1)a_{i,j}(\tau)\chi^i\gamma^j, \sum_{i=0}^{N}\sum_{j=0}^{N-1}(j+1)a_{i,j}(\tau)\chi^i\gamma^j \right)$$
$$-D\left(\sum_{i=0}^{N-2}\sum_{j=0}^{N}(i+2)(i+1)a_{i,j}(\tau)\chi^i\gamma^j + \sum_{i=0}^{N}\sum_{j=0}^{N-2}(j+2)(j+1)a_{i,j}(\tau)\chi^i\gamma^j \right) = 0$$

where $\dot{a}_{i,j}$ is the temporal derivative of $a_{i,j}$. Fixing N to a specific value in the above equation determines the theoretical order of accuracy of the resulting scheme. We set $N = 2$ as in [6] and the resulting

equation is satisfied in the spatial subdomain $[x_{i-1}, x_{i+1}] \times [y_{j-1}, y_{j+1}]$ under the following conditions:

$$\dot{a}_{1,1}(\tau)\chi\gamma = 0 \tag{2.1}$$
$$\dot{a}_{2,0}(\tau)\chi^2 = 0 \tag{2.2}$$
$$\dot{a}_{0,2}(\tau)\gamma^2 = 0 \tag{2.3}$$
$$\dot{a}_{1,0}(\tau)\chi + va_{1,1}(\tau)\chi + 2ua_{2,0}(\tau)\chi = 0 \tag{2.4}$$
$$\dot{a}_{0,1}(\tau)\gamma + ua_{1,1}(\tau)\gamma + 2va_{0,2}(\tau)\gamma = 0 \tag{2.5}$$
$$\dot{a}_{0,0}(\tau) + ua_{1,0}(\tau) + va_{0,1}(\tau) - 2Da_{2,0}(\tau) - 2Da_{0,2}(\tau) = 0 \tag{2.6}$$

The next step consists in fully satisfying the time-varying convection-diffusion equation by proceeding to an analytical integration of the $\dot{a}(\tau)$ function over time. This is done by first integrating the second order terms (2.1), (2.2), (2.3) and substituting the solution in (2.5) and (2.4) to solve for the first order terms $a_{1,0}$ and $a_{0,1}$ which are in turn used to solve the zero-th order term $a_{0,0}$ in equation (2.6) which finally leads to the following solutions:

$$\begin{aligned}
a_{1,1}(\tau) &= a_{1,1}(0) \\
a_{2,0}(\tau) &= a_{2,0}(0) \\
a_{0,2}(\tau) &= a_{0,2}(0) \\
a_{1,0}(\tau) &= -v\tau a_{1,1}(0) - 2u\tau a_{2,0}(0) + a_{1,0}(0) \\
a_{0,1}(\tau) &= -u\tau a_{1,1}(0) - 2v\tau a_{0,2}(0) + a_{0,1}(0) \\
a_{0,0}(\tau) &= uva_{1,1}(0)\tau^2 + u^2 a_{2,0}(0)\tau^2 + v^2 a_{0,2}(0)\tau^2 - ua_{1,0}(0) \\
&\quad - va_{0,1}(0) + 2Da_{2,0}(0)\tau + 2Da_{0,2}(0)\tau + a_{0,0}(0).
\end{aligned}$$

By substituting the $a_{i,j}(\tau)$ coefficients back into the Taylor series expansion (1.2), we obtain an approximation of $\phi(\chi, \gamma, \tau)$:

$$\begin{aligned}
\phi(\chi,\gamma,\tau) \;=\; & a_{0,0} + a_{1,0}(0)(\chi - u\tau) + a_{0,1}(0)(\gamma - v\tau) \\
& + a_{1,1}(0)(uv\tau^2 - v\tau\chi - u\tau\gamma + \chi\gamma) \\
& + a_{2,0}(0)(u^2\tau^2 + 2D\tau - 2u\tau\chi + \chi^2) \\
& + a_{0,2}(0)(v^2\tau^2 + 2D\tau - 2v\tau\gamma + \gamma^2).
\end{aligned} \tag{2.7}$$

In order to obtain a non-trivial solution (different than $\phi_{i,j}^{n+1} = \phi_{i,j}^{n}$) and a resulting method that is symmetric, the cross terms $\tau\gamma$, $\tau\chi$ and the $a_{1,1}$ terms were removed from the previous equation[2]. Keeping $a_{1,1}$ leads to asymmetries in the resulting method and was experimentally shown not to remain unconditionally stable. Removing these terms is essential to obtain a non-trivial solution but may affect the true order of accuracy of the method. Nonetheless, the resulting method is shown empirically to be

[2]This is when symbolic computation becomes considerably useful, choosing to remove $a_{1,1}$ was found by empirically eliminating terms and testing the numerical method with large simulation steps. If the $a_{1,1}$ is kept, the resulting method is only conditionally stable.

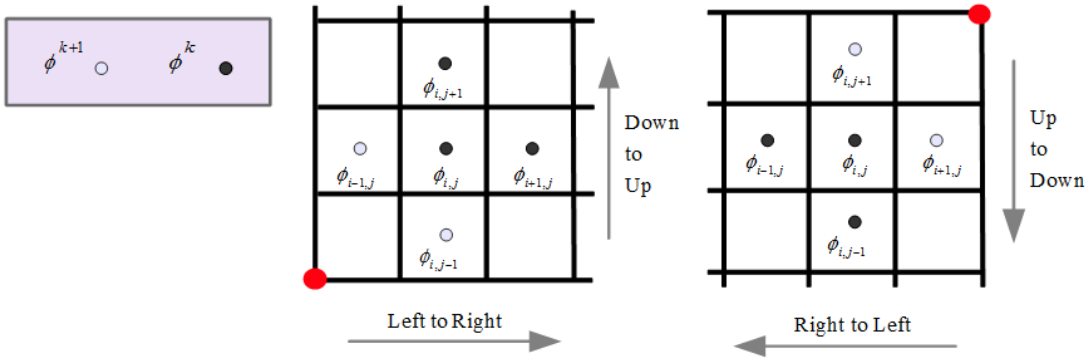

Figure 2.1: Alternating direction

of unconditional stability and resembles greatly the one-dimensional formulas given in [6].

$$\begin{aligned}\phi(\chi,\gamma,\tau) &= a_{0,0} + a_{1,0}(0)(\chi - u\tau) + a_{0,1}(0)(\gamma - v\tau) \\ &\quad + a_{2,0}(0)(u^2\tau^2 + 2D\tau + \chi^2) \\ &\quad + a_{0,2}(0)(v^2\tau^2 + 2D\tau + \gamma^2)\end{aligned} \tag{2.8}$$

Setting the change of variables to $\chi = \gamma = 0$ and $\tau = \Delta t$ leads to a temporal update formula of the solution from $\phi_{i,j}^k$ to $\phi_{i,j}^{k+1}$:

$$\begin{aligned}\phi_{i,j}^{k+1} &= \phi_{i,j}^k - a_{1,0}(0)u\Delta t - a_{0,1}(0)v\Delta t \\ &\quad + a_{2,0}(0)(u^2\Delta t^2 + 2D\Delta t) + a_{0,2}(0)(v^2\Delta t^2 + 2D\Delta t) \quad \text{(Update Formula)}\end{aligned}$$

As we can see from this equation, there are four unknown coefficients $a_{1,0}, a_{0,1}, a_{2,0}, a_{0,2}$ which need to be mapped to samples on the grid around $\phi_{i,j}$. By judiciously choosing elements from the current sample's neighborhood, it is possible to form a system of 4 equations which once solved gives a correct mapping of the $a_{1,0}, a_{0,1}, a_{2,0}, a_{0,2}$ coefficients. A particular configuration of elements will result in formula which solves the considered equation and propagates the boundary information into the solution from one side to the other. Hence, it is required to balance the method by proposing two traversal directions; left-to-right and right-to-left.

2.2 Left-to-Right Algorithm

For the left to right algorithm, we select the samples illustrated in figure (2.1) and effectuate their corresponding substitutions of χ, γ and τ:

$$\begin{array}{lll} \phi^k_{i+1,j}: & \chi = \Delta x, & \gamma = 0, \quad \tau = 0 \\ \phi^k_{i,j+1}: & \chi = 0, & \gamma = \Delta y, \quad \tau = 0 \\ \phi^{k+1}_{i-1,j}: & \chi = -\Delta x, & \gamma = 0, \quad \tau = \Delta t \\ \phi^{k+1}_{i,j-1}: & \chi = 0, & \gamma = -\Delta y, \quad \tau = \Delta t \end{array}$$

The above substitutions into (2.8) leads to a system of four equations and four unknowns from which we can form a linear system $Ax = b$:

$$\left(\begin{array}{cccc|c} \Delta x & 0 & \Delta x^2 & 0 & a_{1,0}(0) \\ 0 & \Delta y & 0 & \Delta y^2 & a_{0,1}(0) \\ -\Delta x - u\Delta t & -v\Delta t & u^2\Delta t^2 + 2D\Delta t + \Delta x^2 & v^2\Delta t^2 + 2D\Delta t & a_{2,0}(0) \\ -u\Delta t & -\Delta y - v\Delta t & u^2\Delta t^2 + 2D\Delta t & v^2\Delta t^2 + 2D\Delta t + \Delta y^2 & a_{0,2}(0) \end{array}\right) = \left(\begin{array}{c} \phi^k_{i+1,j} - \phi^k_{i,j} \\ \phi^k_{i,j+1} - \phi^k_{i,j} \\ \phi^{k+1}_{i-1,j} - \phi^k_{i,j} \\ \phi^{k+1}_{i,j-1} - \phi^k_{i,j} \end{array}\right)$$

A change in variables according to the update equation (2.9) yields:

$$\begin{array}{rcl} a_{1,0}(0) & \rightarrow & -(u\Delta t)^{-1}\bar{a}_{1,0}(0) \\ a_{0,1}(0) & \rightarrow & -(v\Delta t)^{-1}\bar{a}_{0,1}(0) \\ a_{2,0}(0) & \rightarrow & (u^2\Delta t^2 - 2D\Delta t)^{-1}\bar{a}_{2,0}(0) \\ a_{0,2}(0) & \rightarrow & (v^2\Delta t^2 - 2D\Delta t)^{-1}\bar{a}_{0,2}(0) \end{array} \qquad \text{(Change in variables)}$$

and the following system is obtained:

$$\left(\begin{array}{cccc|c} -{c_1}^{-1} & 0 & (c_1^2 + 2r_1)^{-1} & 0 & \bar{a}_{1,0}(0) \\ 0 & -{c_2}^{-1} & 0 & (c_2^2 + 2r_2)^{-1} & \bar{a}_{0,1}(0) \\ 1 + {c_1}^{-1} & 1 & 1 + (c_1^2 + 2r_1)^{-1} & 1 & \bar{a}_{2,0}(0) \\ 1 & 1 + {c_2}^{-1} & 1 & 1 + (c_2^2 + 2r_2)^{-1} & \bar{a}_{0,2}(0) \end{array}\right) = \left(\begin{array}{c} \phi^k_{i+1,j} - \phi^k_{i,j} \\ \phi^k_{i,j+1} - \phi^k_{i,j} \\ \phi^{k+1}_{i-1,j} - \phi^k_{i,j} \\ \phi^{k+1}_{i,j-1} - \phi^k_{i,j} \end{array}\right)$$

where $\mathbf{c} = \frac{\mathbf{u}\Delta t}{\Delta \mathbf{x}}$ is the 2-dimensional Courant number and $\mathbf{r} = \frac{D\Delta t}{(\Delta \mathbf{x})^2}$ a 2-dimensional diffusion coefficient. The above system was solved with the symbolic language Mathematica and the following solution was obtained:

$$\begin{aligned} \phi^{k+1}_{i,j} = {} & \frac{(-c_1 + c_1^2 + 2r_1)\phi^k_{i+1,j}}{2 + c_1 + c_1^2 + c_2 + c_2^2 + 2r_1 + 2r_2} + \frac{(c_1 + c_1^2 + 2r_1)\phi^{k+1}_{i-1,j}}{2 + c_1 + c_1^2 + c_2 + c_2^2 + 2r_1 + 2r_2} \\ & + \frac{(-c_2 + c_2^2 + 2r_2)\phi^k_{i,j+1}}{2 + c_1 + c_1^2 + c_2 + c_2^2 + 2r_1 + 2r_2} + \frac{(c_2 + c_2^2 + 2r_2)\phi^{k+1}_{i,j-1}}{2 + c_1 + c_1^2 + c_2 + c_2^2 + 2r_1 + 2r_2} \\ & + \frac{(2 + c_1 - c_1^2 + c_2 - c_2^2 - 2r_1 - 2r_2)\phi^k_{i,j}}{2 + c_1 + c_1^2 + c_2 + c_2^2 + 2r_1 + 2r_2} \end{aligned} \qquad (2.9)$$

It is noticeable how this formula resembles the expression obtained in [6]. The scale denominator and the coefficient in front of the $\phi^k_{i,j}$ term are augmented. Such an outline leads to insight on the natural extension to higher dimensions than 2 and this topic is covered later.

As mentioned earlier, the formula (2.10) is used to solve the equation while traversing the grid from left to right and therefore only propagates the boundary information from one side to the other. In order to equilibrate the method, an other traversal direction is proposed right-to-left.

2.3 Right-to-Left Algorithm

For the right-to-left algorithm (2.10), we select the samples illustrated in figure (2.1) and perform their corresponding change in variable for χ, γ and τ:

$$\begin{array}{lllll}
\phi^k_{i-1,j}: & \chi = -\Delta x, & \gamma = 0, & \tau = 0 \\
\phi^k_{i,j-1}: & \chi = 0, & \gamma = -\Delta y, & \tau = 0 \\
\phi^{k+1}_{i+1,j}: & \chi = \Delta x, & \gamma = 0, & \tau = \Delta t \\
\phi^{k+1}_{i,j+1}: & \chi = 0, & \gamma = \Delta y, & \tau = \Delta t
\end{array}$$

By proceeding with an analogous development as for the left-to-right algorithm, we obtain the following closed form formula to solve the computational cells from right to left:

$$\begin{aligned}
\phi^{k+1}_{i,j} = &\frac{(-c_1 + c_1^2 + 2r_1)\phi^{k+1}_{i+1,j}}{2 - c_1 + c_1^2 - c_2 + c_2^2 + 2r_1 + 2r_2} + \frac{(c_1 + c_1^2 + 2r_1)\phi^{k}_{i-1,j}}{2 - c_1 + c_1^2 - c_2 + c_2^2 + 2r_1 + 2r_2} \\
&+ \frac{(-c_2 + c_2^2 + 2r_2)\phi^{k+1}_{i,j+1}}{2 - c_1 + c_1^2 - c_2 + c_2^2 + 2r_1 + 2r_2} + \frac{(c_2 + c_2^2 + 2r_2)\phi^{k}_{i,j-1}}{2 - c_1 + c_1^2 - c_2 + c_2^2 + 2r_1 + 2r_2} \\
&+ \frac{(2 - c_1 - c_1^2 - c_2 - c_2^2 - 2r_1 - 2r_2)\phi^{k}_{i,j}}{2 - c_1 + c_1^2 - c_2 + c_2^2 + 2r_1 + 2r_2}
\end{aligned} \tag{2.10}$$

2.4 ADE Algorithm

In order to correctly solve the unsteady convection-diffusion equation with the proposed method, it is first required to solve the computational grid following one direction; for instance the left-to-right direction (2.10), then alternate the direction with the right-to-left traversal (2.11) and vice versa as illustrated in figure 2.1.

3 Numerical Simulations

As the main features of the method are unconditional stability and ease of implementation, the experiments conducted in this section are mainly illustrations of the method's capacity to solve the convection-diffusion equation rather than achieving high precision. Note that unconditional stability is not proved mathematically in this text but is evident when empirically testing for stability by running simulations with very large time steps. Time steps with which explicit integration methods would normally lead to numerical explosions. Such an experiment is also provided on this web page : (Deprecated, send e-mail to first author)`http://pages.usherbrooke.ca/mguay/ADE`, along the numerical simulations of this section.

In particular we considered in this paper, two initial boundary problems both simulated over a square grid. The first one formulated with the unsteady convection-diffusion equation and the second with the unsteady diffusion equation as mean of outlining the performance with diffusion dominated problems.

3.1 Unsteady Convection-Diffusion Equation Problem

Let us consider the first initial boundary problem by referring to equation (1.1) and setting the convection vector to $\mathbf{u} = \{1, 1\}$ and the diffusion coefficient to $D = 1$

$$\frac{\partial \phi}{\partial t}(\mathbf{x}, t) + \frac{\partial \phi}{\partial x}(\mathbf{x}, t) + \frac{\partial \phi}{\partial y}(\mathbf{x}, t) - \frac{\partial^2 \phi}{\partial x^2}(\mathbf{x}, t) - \frac{\partial^2 \phi}{\partial y^2}(\mathbf{x}, t) = 0, \tag{3.1}$$

considering boundary conditions:

$$\begin{cases} \phi(0, y, t) = \frac{1}{4t+1} \exp\left[-\frac{(-0.05-t)^2}{4t+1} - \frac{(y-0.05-t)^2}{4t+1}\right] & \text{if } x = 0, \\ \phi(1, y, t) = \frac{1}{4t+1} \exp\left[-\frac{(0.95-t)^2}{4t+1} - \frac{(y-0.05-t)^2}{4t+1}\right] & \text{if } x = 1, \\ \phi(x, 0, t) = \frac{1}{4t+1} \exp\left[-\frac{(x-0.05-t)^2}{4t+1} - \frac{(-0.05-t)^2}{4t+1}\right] & \text{if } y = 0, \\ \phi(x, 1, t) = \frac{1}{4t+1} \exp\left[-\frac{(x-0.05-t)^2}{4t+1} - \frac{(0.95-t)^2}{4t+1}\right] & \text{if } y = 1 \end{cases}$$

and initializing the solution at time $t = 0$ with:

$$\phi(\mathbf{x}, 0) = \exp\left[-(x - 0.05)^2 - (y - 0.05)^2\right].$$

The simulation consists of 1000 iterations of time step $\Delta t = 0.001$ forming $T = 1$ second, the grid holds 40 cells of spatial resolution $\Delta \mathbf{x} = \left\{\frac{1}{40}, \frac{1}{40}\right\}$ forming the spatial subdomain $(0, 1) \times (0, 1)$. The final solution is illustrated in figure 3.2 and can be compared with the exact solution:

$$\phi(\mathbf{x}, t) = \frac{1}{4t + 1} \exp\left[-\frac{(x - 0.05 - t)^2}{4t + 1} - \frac{(y - 0.05 - t)^2}{4t + 1}\right] \tag{3.2}$$

which is also illustrated in figure 3.4.

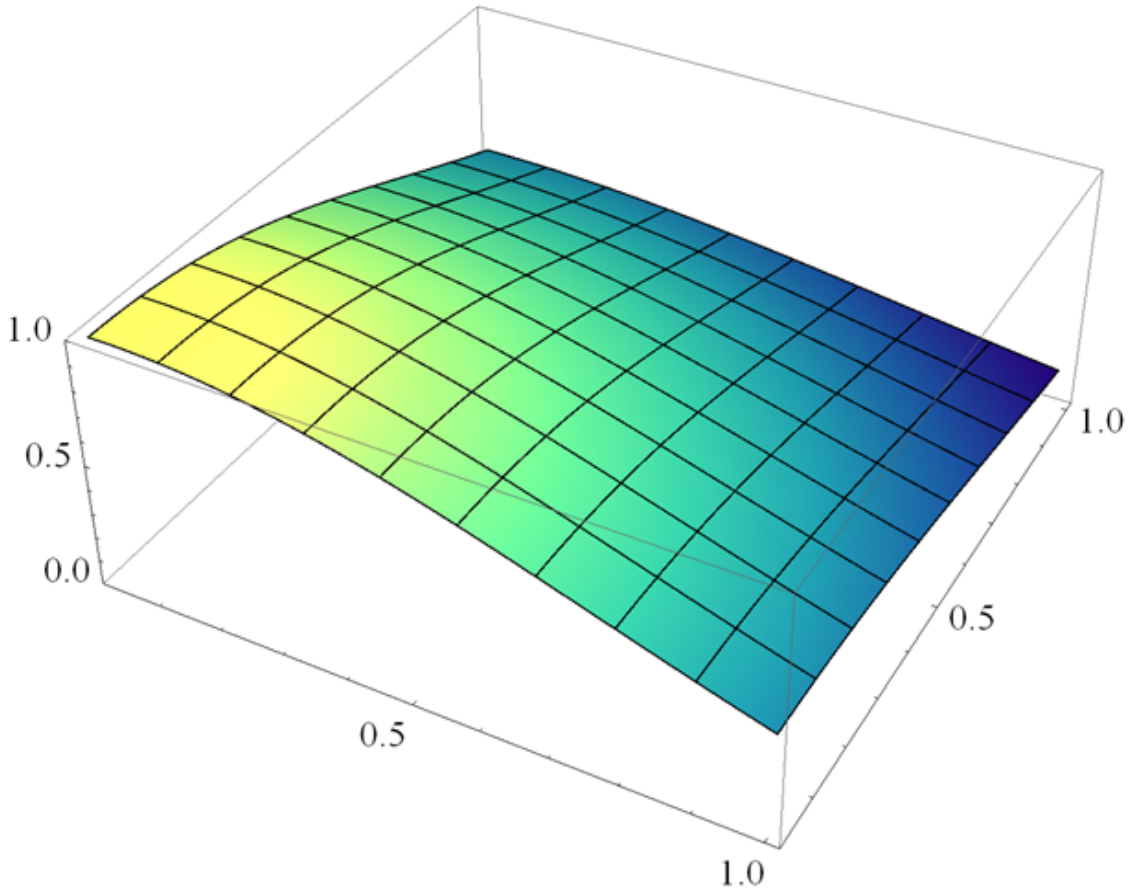

Figure 3.1: Initial solution to the initial-boundary problem (3.1) at time $T = 0$.

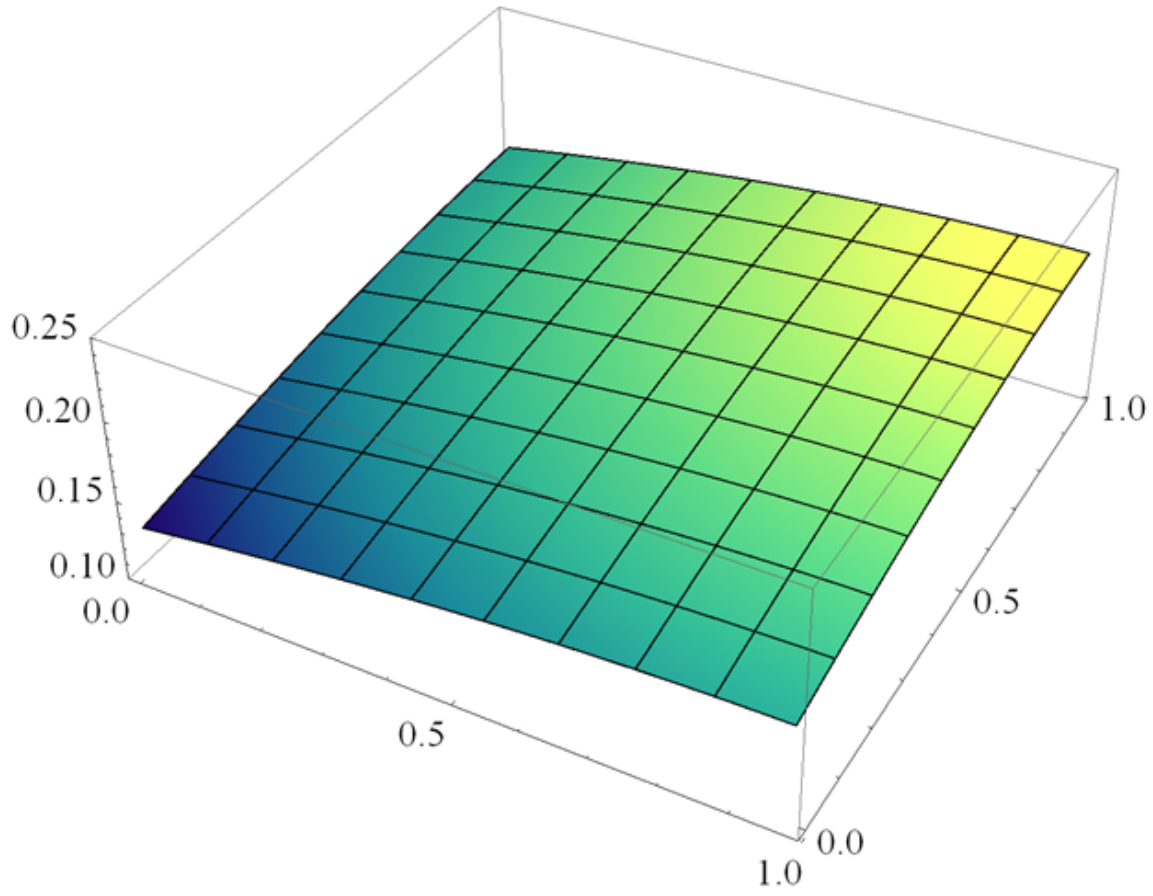

Figure 3.2: Numerical approximation to a solution of the initial-boundary problem (3.1) after $T = 1$ second of simulation time with $\Delta t = 0.001$ and $\Delta \mathbf{x} = \left\{\frac{1}{40}, \frac{1}{40}\right\}$.

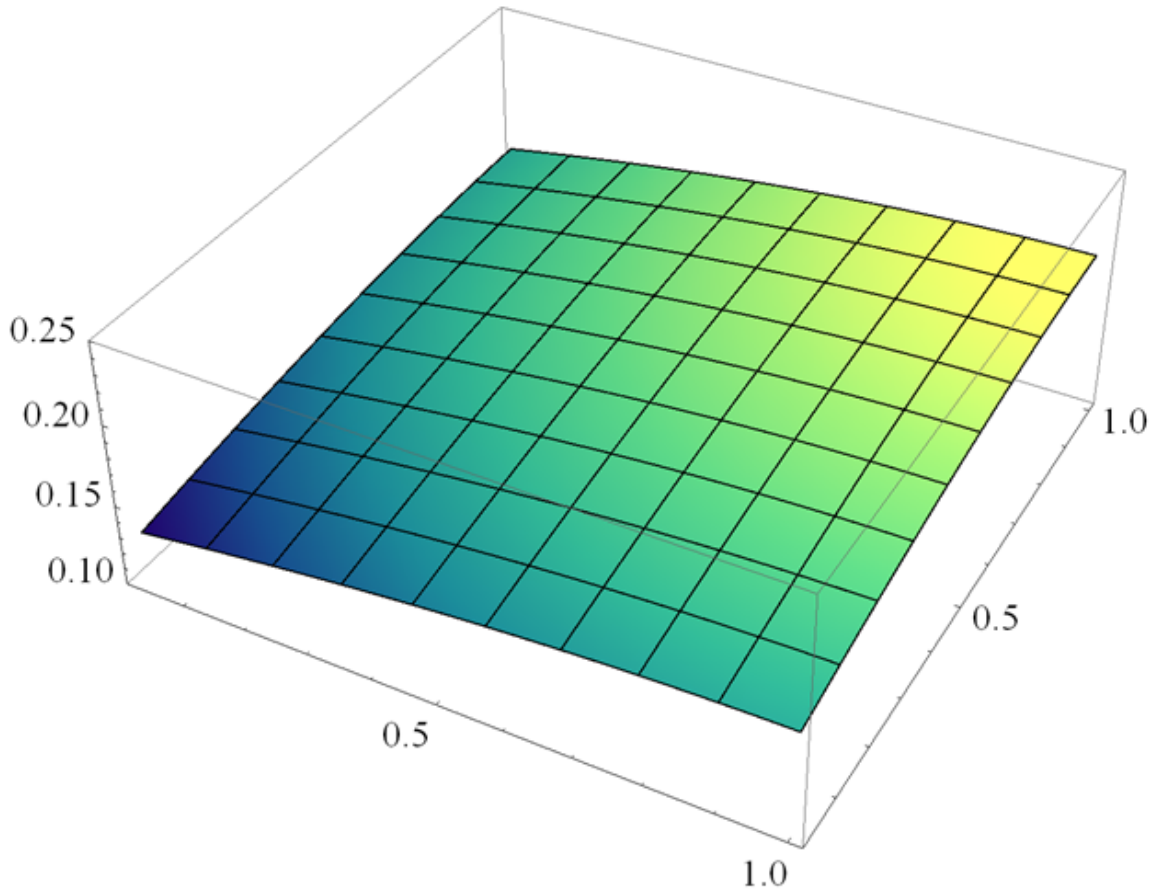

Figure 3.3: Exact solution of the initial-boundary problem (3.1) at time $T = 1$.

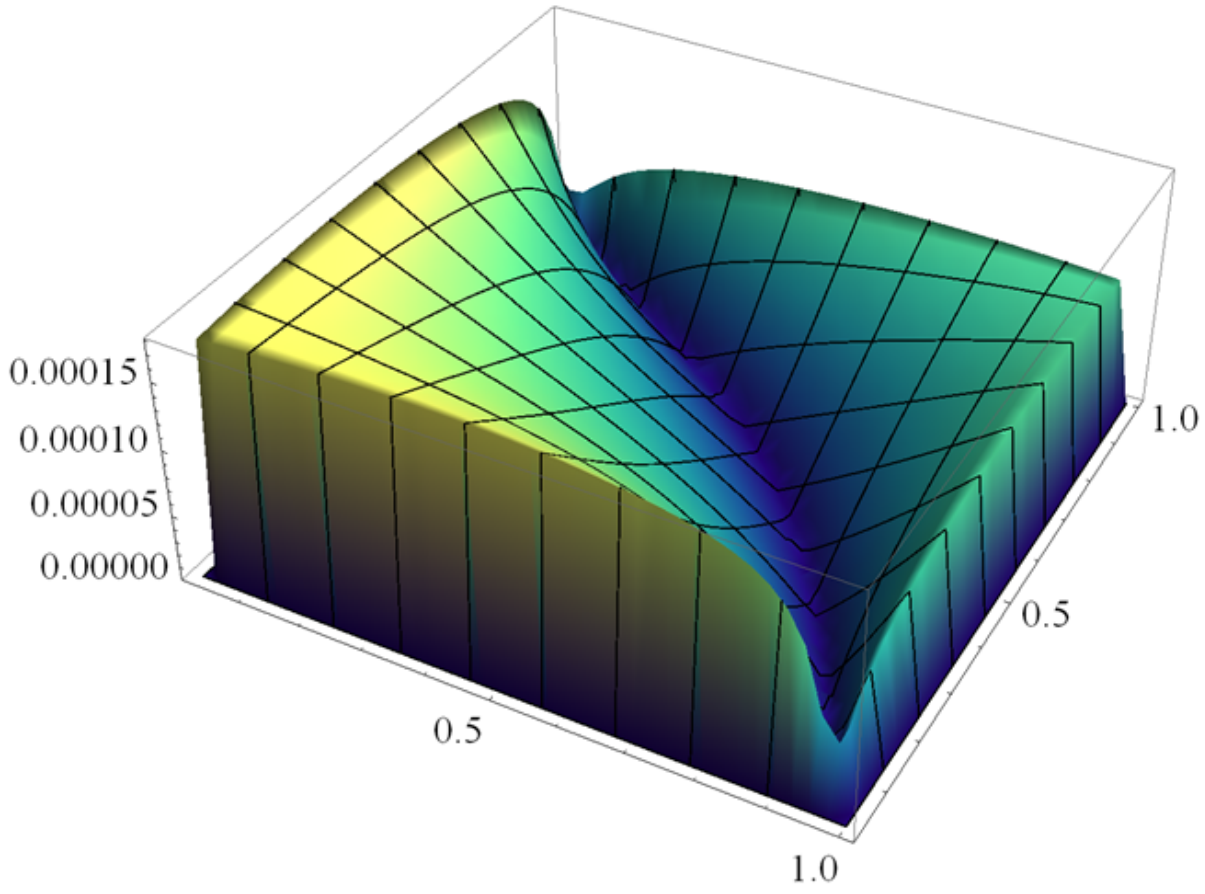

Figure 3.4: Absolute error between exact and computed solutions of the initial-boundary problem (3.1) after $T = 1$ second of simulation with $\Delta t = 0.001$ and $\Delta \mathbf{x} = \left\{\frac{1}{40}, \frac{1}{40}\right\}$. Maximum error is 1.81677×10^{-4}.

3.2 Unsteady Diffusion Equation Problem

Although the proposed method is developed to solve the unsteady convection-diffusion equation, the behavior of the method when solving diffusion dominated problems is illustrated by solving the 2-dimensional unsteady diffusion equation subject to Dirichlet boundary conditions over a square grid. An initial boundary problem is formulated by setting $\mathbf{c} = \{0, 0\}$, $D = \frac{1}{\pi^2}$ in equation (1.1) resulting in

$$\frac{\partial \phi}{\partial t}(\mathbf{x}, t) = \frac{1}{\pi^2}\left(\frac{\partial^2 \phi}{\partial x^2}(\mathbf{x}, t) + \frac{\partial^2 \phi}{\partial y^2}(\mathbf{x}, t)\right), \tag{3.3}$$

considering boundary conditions:

$$\begin{cases} \phi(x, 0, t) = e^{-\frac{t}{2}} \cos((x)\frac{\pi}{2}) + e^{-2t} \sin(x\pi), \\ \phi(x, 1, t) = e^{-\frac{t}{2}} \cos((x+1)\frac{\pi}{2}) + e^{-2t} \sin((x-1)\pi), \\ \phi(0, y, t) = e^{-\frac{t}{2}} \cos((y)\frac{\pi}{2}) + e^{-2t} \sin(y\pi), \\ \phi(1, y, t) = e^{-\frac{t}{2}} \cos((1+y)\frac{\pi}{2}) + e^{-2t} \sin((1-y)\pi) \end{cases}$$

and initializing the solution at time $t = 0$ with:

$$\phi(\mathbf{x}, 0) = cos((x+y)\frac{\pi}{2}) + sin((x-y)\pi)$$

and is simulated for 20 iterations of temporal resolution $\Delta t = 0.05$ forming $T = 1$ second of simulation and the final solution is illustrated in figure 3.6. The approximated solution can be compared with the exact solution

$$\phi(\mathbf{x}, t) = e^{-\frac{t}{2}} cos((x+y)\frac{\pi}{2}) + e^{-2t} sin((x-y)\pi) \tag{3.4}$$

which is also illustrated in figure 3.7.

4 Further Extensions of the Method

It is straight forward to extend the formulas obtained for the left-to-right (2.10) and right-to-left (2.11) directions to three or more dimensions by comparing them to the one-dimensional formulas obtained in Xie [6]. A generalization of the left to right algorithm is given by:

$$\begin{aligned} \phi_{\mathbf{i}}^{k+1} = \quad & \frac{\left(2 + \sum_{j=1}^{N} (c_j - c_j^2 - 2r_j)\right) \phi_{\mathbf{i}}^{k}}{2 + \sum_{j=1}^{N} (c_j + c_j^2 + 2r_j)} + \\ & \sum_{j=1}^{N} \frac{\left((2r_j + c_j^2 + c_j)\phi_{\mathbf{i}-\delta(j)}^{k+1} + (2r_j + c_j^2 - c_j)\phi_{\mathbf{i}+\delta(j)}^{k}\right)}{2 + \sum_{j=1}^{N} (c_j + c_j^2 + 2r_j)} \end{aligned}$$

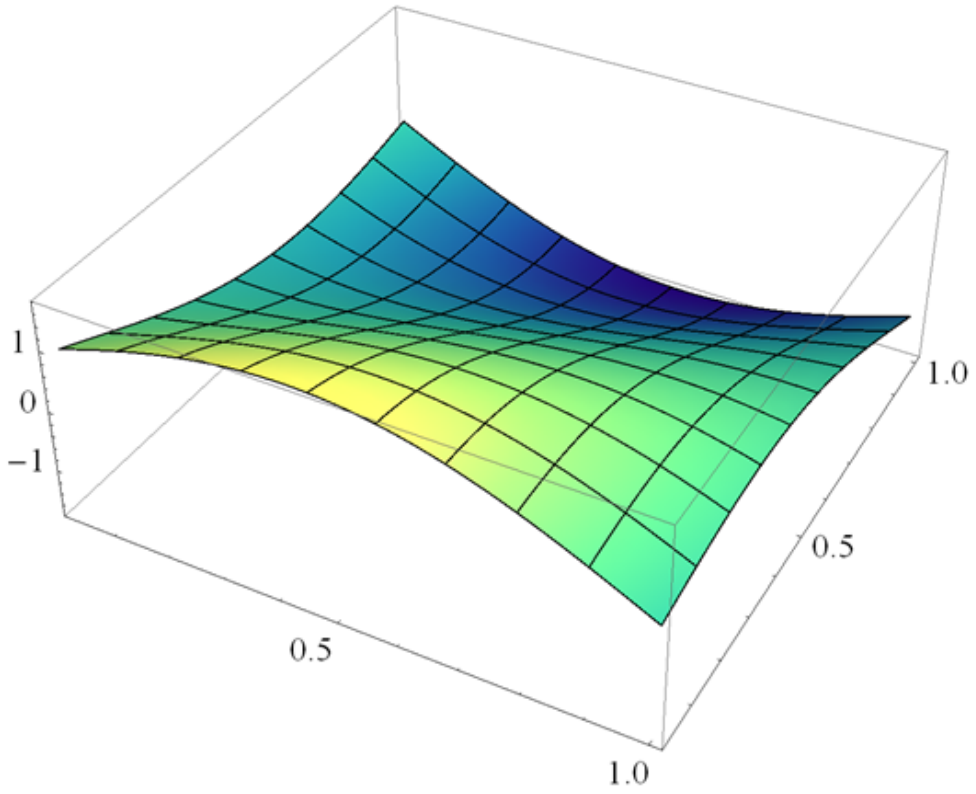

Figure 3.5: Initial solution of the initial-boundary problem (3.3) at time $T = 0$.

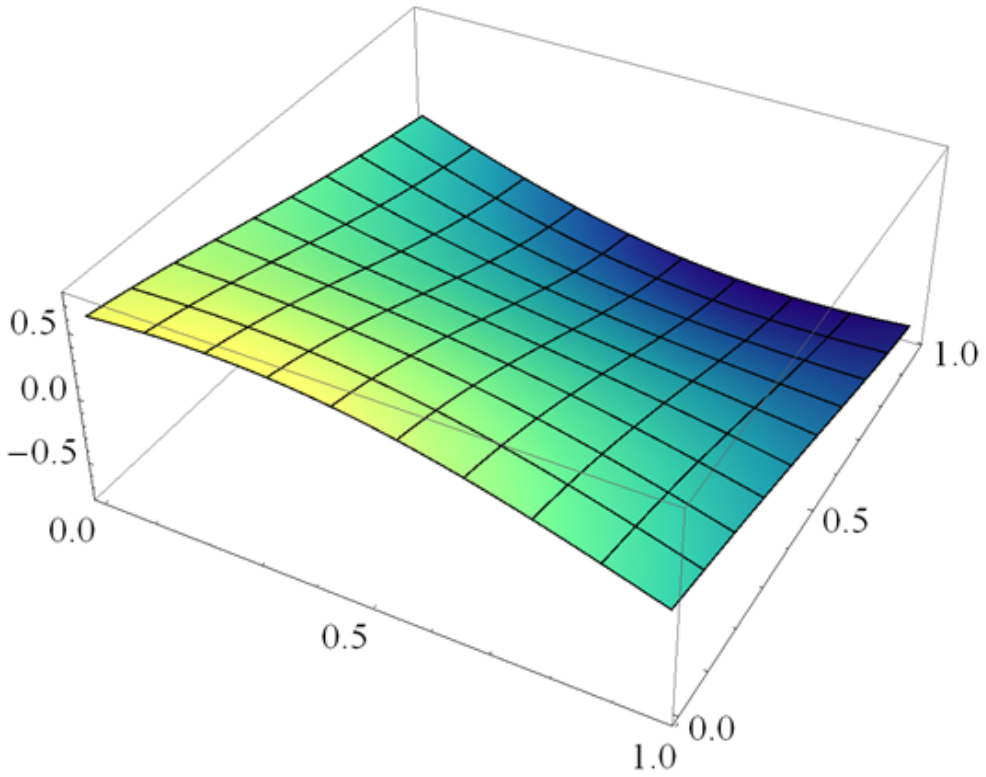

Figure 3.6: Numerical approximation of a solution of the initial-boundary problem (3.3) after $T = 1$ second of simulation with $\Delta t = 0.001$ and $\Delta \mathbf{x} = \left\{\frac{1}{40}, \frac{1}{40}\right\}$.

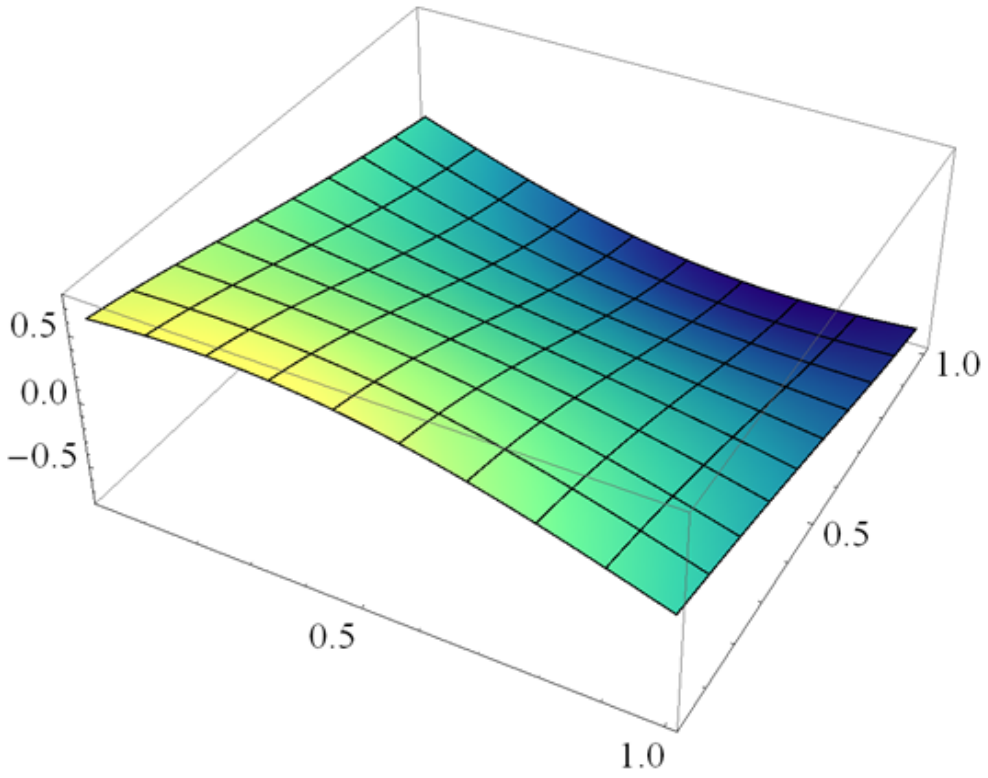

Figure 3.7: Exact solution of the initial-boundary problem (3.3) at time $T = 1$.

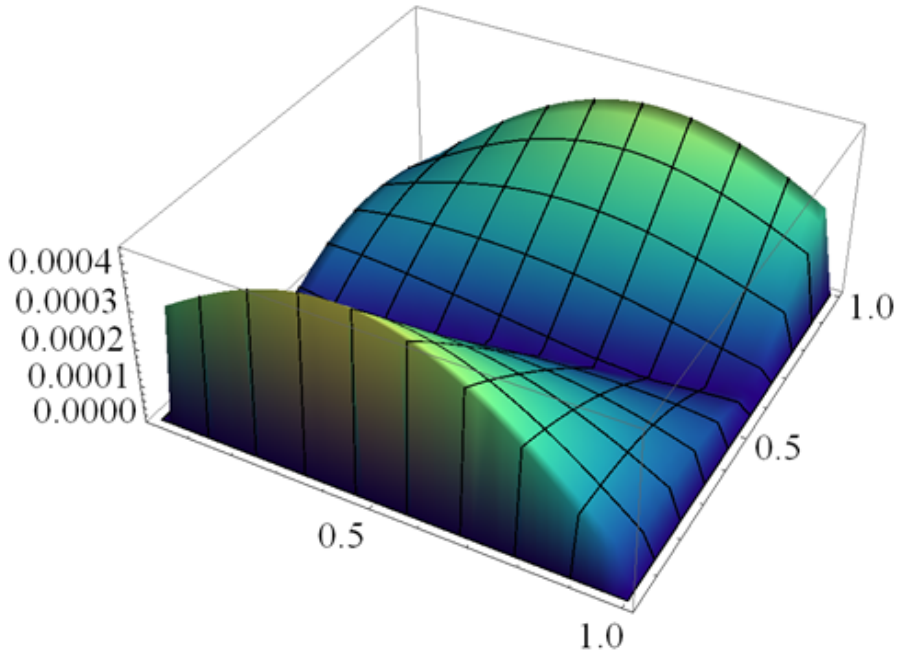

Figure 3.8: Absolute error between exact and computed solutions of the initial-boundary problem (3.3) after $T = 1$ second of simulation with $\Delta t = 0.001$ and $\Delta \mathbf{x} = \left\{\frac{1}{40}, \frac{1}{40}\right\}$. Maximum error is 4.60674×10^{-4}.

and the right to left algorithm is given by:

$$\phi_{\mathbf{i}}^{k+1} = \frac{\left(2 + \sum_{j=1}^{N}\left(-c_j - c_j^2 - 2r_j\right)\right)\phi_{\mathbf{i}}^{k}}{2 + \sum_{j=1}^{N}\left(-c_j + c_j^2 + 2r_j\right)} + \sum_{j=1}^{N} \frac{\left((2r_j + c_j^2 + c_j)\phi_{\mathbf{i}-\delta(j)}^{k+1} + (2r_j + c_j^2 - c_j)\phi_{\mathbf{i}+\delta(j)}^{k}\right)}{2 + \sum_{j=1}^{N}\left(-c_j + c_j^2 + 2r_j\right)}$$

where $\delta(j) = (0, ..., 1, ..., 0)$ and the index of the non-null component is j.

Unfortunately, extensions to higher orders of accuracy for both space and time are not trivial and the same issue arises when applying the method to other equations than the one considered in this paper such as non-linear equations or systems of dependent partial differential equations such as the Navier-Stokes equations. We can see from the derivation of the two-dimensional case that the complexity of the system to solve increases with the number of elements, or unknowns, considered in the local series expansion. Therefore, it becomes quite challenging to solve such a system symbolically. However, this could be addressed by making use of sophisticated techniques to symbolically solve the type of systems encountered when extending to higher orders of accuracy or other classes of partial differential equations, such a solver could extract a symbolic solution more efficiently.

5 Conclusion

Extensions to higher dimensions of the method proposed by [6] were illustrated. Its main features are explicit traversal, unconditional stability (empirical demonstration) and ease of implementation. The capacity of the proposed method to solve the time-varying convection-diffusion equation in two dimensions was demonstrated and links ((Deprecated, send e-mail to first author)`http://pages.usherbrooke.ca/mguay/ADE`) to the derivation code as well as stability tests were provided in order to empirically show the unconditional stability of the method. Finally a small discussion on future potential outcomes concerning the extensions to higher orders of accuracy or to other classes of partial differential equations were given.

References

[1] D.J. Evans and A.R. Abdullah. A new explicit method for the diffusion-convection equation. *Computers and Mathematics with Applications*, 11(1-3):145–154, 1985.

[2] A.R. Gourlay. Hopscotch: a fast second-order partial differential equation solver. *IMA Journal of Applied Mathematics*, 6(4):375–390, 1970.

[3] S. Leung and S. Oshery. An alternating direction explicit (ade) scheme for time-dependent evolution equations. *Preprint UCLA June 9, 2005*, 2005.

[4] V.K. Saul'yev. Integration of equations of parabolic type by the method of nets. *Pregamon Press*, pages 121–128, 1964.

[5] A. Selle, R. Fedkiw, K. Byungmoon, L. Yingjie, and R. Jarek. An unconditionally stable maccormack method. *Journal of Scientific Computing*, 35(2-3):350–371, 2007.

[6] J. Xie, Z. Lin and J. Zhou. A new unconditionally stable explicit scheme for the convection-diffusion equation with robin boundary conditions. *International Journal of Computer Mathematics*, 85(12): 1833–1847, 2008.

[7] B.L. Zhang and X.M. Su. Alternating block explicit-implicit method for the two-dimensional diffusion equation. *International Journal of Computationer Mathematics*, 38(3-4):241–255, 2001.