Methods and Applications of Analysis.
Vol. 26, No. 3, pp. 249–268, September 2019
© 2019 International Press
003

AN ALTERNATING DIRECTION EXPLICIT METHOD FOR TIME EVOLUTION EQUATIONS WITH APPLICATIONS TO FRACTIONAL DIFFERENTIAL EQUATIONS*

HAO LIU† AND SHINGYU LEUNG‡

Dedicated to Professor Roland Glowinski on the occasion of his 80th birthday

Abstract. We derive and analyze the alternating direction explicit (ADE) method for time evolution equations with the time-dependent Dirichlet boundary condition and with the zero Neumann boundary condition. The original ADE method is an additive operator splitting (AOS) method, which has been developed for treating a wide range of linear and nonlinear time evolution equations with the zero Dirichlet boundary condition. For linear equations, it has been shown to achieve the second order accuracy in time yet is unconditionally stable for an arbitrary time step size. For the boundary conditions considered in this work, we carefully construct the updating formula at grid points near the boundary of the computational domain and show that these formulas maintain the desired accuracy and the property of unconditional stability. We also construct numerical methods based on the ADE scheme for two classes of fractional differential equations. We will give numerical examples to demonstrate the simplicity and the computational efficiency of the method.

Key words. Numerical Methods, Partial Differential Equations, Fractional Derivatives, Operator Splitting Methods.

Mathematics Subject Classification. 65M06, 65M12, 26A33.

1. Introduction. Time evolution equations are widely used in modeling physical phenomenon and chemical reactions. It is therefore important to develop efficient and accurate numerical schemes to approximate the solution. This paper considers the approximation of the evolution of some quantity $u(\mathbf{x}, t)$ on a bounded domain Ω satisfying the following initial value problem

$$\begin{cases} \frac{\partial u}{\partial t} = \mathcal{L}(u, t)\,, \\ u(\mathbf{x}, 0) = u_0(\mathbf{x}). \end{cases} \tag{1}$$

There are various finite difference schemes developed according to the choice of the differential operator $\mathcal{L}$ [32]. The simplest approach is the explicit method given by

$$u_i^{n+1} = u_i^n + \Delta t \mathcal{L}(u^n, t^n)$$

with $u_i^n = u(\mathbf{x}_i, t^n)$. This scheme is developed based on the classical forward Euler time marching and therefore is at best first order accurate in time. The accuracy in space depends on the space discretization scheme used to approximate the differential operator $\mathcal{L}(u, t)$. Relating to the stability of the method, these methods usually come with a strict restriction on the time step chosen based on the mesh resolution. If the differential operator $\mathcal{L}$ is complicated, the time step used in the numerical scheme needs to be extremely small to maintain the numerical stability.

To relax the stability condition or to obtain a higher order accurate numerical solution, it is usually more favorable to consider the implicit methods or semi-implicit

*Received December 22, 2017; accepted for publication September 19, 2019.
†School of Mathematics, Georgia Institute of Technology, 686 Cherry Street, Atlanta, GA 30332-0160, USA (hao.liu@math.gatech.edu).
‡Department of Mathematics, the Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong (masyleung@ust.hk).

methods. If $\mathcal{L}$ is linear, we can easily find the inverse operator, $(I-\Delta t\mathcal{L})^{-1}$, and can directly update u^{n+1} implicitly by $u^{n+1}=(I-\Delta t\mathcal{L})^{-1}(u^n)$. This implicit scheme is first order accurate in time but unconditionally stable. For a nonlinear operator $\mathcal{L}$, however, it is usually not that straightforward how to obtain an unconditionally stable numerical scheme. One possibility is to follow the regularization approach as discussed in [31, 37]. A Laplacian term has been introduced in [31] as an extra regularization for the initial value problem

$$\begin{cases} \frac{\partial u}{\partial t}=\alpha\Delta u-\alpha\Delta u+\mathcal{L}(u,t)\,, \\ u(\mathbf{x},0)=v(\mathbf{x}). \end{cases}$$

The numerical scheme is

$$\frac{u^{n+1}-u^n}{\Delta t}=\alpha\Delta u^{n+1}-\alpha\Delta u^n+\mathcal{L}(u^n,t^n)\,.$$

This scheme is first order in time and is unconditionally stable. But the choice of α is in general an open problem. Following a similar idea, [37] has introduced an extra curvature term in the level set equation for fast evolution of some curvature dependent flows. We refer all readers to the references and thereafter for a more detailed discussion of the approach.

Other than the regularization technique, another possible approach is to linearize the nonlinear operator $\mathcal{L}$ by $L(u^n)u$ and obtains the solution at the time level $t=t_{n+1}$. The Crank-Nicolson scheme is a famous approach to this linearized equation and the method is written as

$$u^{n+1}=\left(I-\frac{\Delta t}{2}L\right)^{-1}\left(I+\frac{\Delta t}{2}L\right)u^n\,.$$

This scheme, as the average of the implicit scheme and the explicit scheme, is second order accurate in time and unconditionally stable (at least to linear equations). But in high dimensions, it might not be trivial to invert such a matrix $\left(I-\frac{\Delta t}{2}L\right)$. Roughly speaking, there are two general operator splitting approaches developed for inverting the operator for a wide range of applications [9, 10, 11, 13, 12]. The first group is the multiplicative operator splitting (MOS) methods which splits the complicated operator into a product of some simple operators so that each of these subproblems can be easily solved. Mathematically, we have

$$u^{n+1}=\prod_{k=1}^{K}\left[I-\Delta tL_k(u^n)\right]^{-1}u^n\,.$$

One famous example of the MOS methods is the alternating direction implicit scheme (ADI). For example, when $Lu=\Delta u$, the ADI scheme reads as

$$\begin{cases} \frac{2}{\Delta t}\left(u^{n+\frac{1}{2}}-u^n\right)=\delta_{xx}u^{n+\frac{1}{2}}+\delta_{yy}u^n\,, \\ \frac{2}{\Delta t}\left(u^{n+1}-u^{n+\frac{1}{2}}\right)=\delta_{xx}u^n+\delta_{yy}u^{n+\frac{1}{2}} \end{cases}$$

where δ_{xx} and δ_{yy} denotes the second derivative operators in the x- and the y-directions, respectively. The ADI method is efficient since the method inverts only a tridiagonal matrix in each sub-step. For two dimensional problems, the ADI scheme

is unconditionally stable. But for higher dimensional problems, the trivial implementation becomes only conditionally stable. Special strategy is needed to keep the unconditional stability.

Another group of splitting method is the additive operator splitting (AOS) methods [25, 26, 35]. Instead of approximating the inverse by a product of simpler operators, the AOS methods split the operator into a summation of some simple operators. Mathematically, we can approximate the time evolution as

$$u^{n+1} = \frac{1}{m} \sum_{k=1}^{K} \left[I - \Delta t L_k(u^n) \right]^{-1} u^n \,.$$

In general, the choice for the splitting might be equation-specific or might not be easy to generalize in high dimensions. One general approach is the alternating direction explicit (ADE) method. The ADE scheme was first introduced and analyzed in [19, 1] for the linear heat equation. It has recently been extended to a much wider class of nonlinear PDEs in [20]. For linear problems, the ADE method can be proved to be second order accurate in both time and space and is unconditionally stable for arbitrary time step size for both low dimensional and high dimensional problems. Using the sweeping strategy in updating the solution sequentially, the ADE scheme can be implemented in a fully explicitly fashion. In [20], the method has been carefully analyzed in the operator form and has been extended to various nonlinear equations including the fourth order diffusion equation, the Hamilton-Jacobi equation, the fourth order nonlinear lubrication equation, and etc. More recently, the method has been coupled with the fast Huygens sweeping method [21] for nematic liquid crystal modeling [17]. These discussions, however, concentrate only on the constant Dirichlet boundary condition. In this work, we analyze the ADE method for the heat equation with the time dependent Dirichlet boundary condition and also the Neumann boundary condition. To demonstrate the effectiveness of the approach, we then apply the ADE scheme to solve two kinds of fractional differential equations.

This paper is organized as follows. In Section 2, we first show some existing results on the ADE method for problems with the constant Dirichlet boundary condition. Then we derive and perform a truncation error analysis of the ADE method for the heat equation with the time-dependent Dirichlet boundary condition and the Neumann boundary condition. In Section 3.1, we apply our ADE method to two kinds of fractional differential equations: the time distributed order super-diffusive equation and the sub-diffusive diffusion-reaction system. Results of numerical experiments are shown in Section 4. Finally in Section 5 we summarize and conclude the ADE method and our results.

2. The alternating direction explicit (ADE) method. In this section, we first give a brief introduction to the ADE scheme for problem with the zero Dirichlet boundary condition. For a more detailed discussion on the numerical scheme, we refer all readers to [20]. Then, we will extend the formulation to problems with the time-dependent Dirichlet boundary condition and also the zero Neumann boundary condition.

2.1. The ADE scheme for linear time dependent PDEs. We consider the following linear time dependent PDE

$$\frac{\partial u}{\partial t} = \mathcal{L}(u) \,,$$

one first discretizes the right hand side to get

$$\frac{\partial \mathbf{u}}{\partial t} = A\mathbf{u} + b$$

where A is the discretization of the linear differential operator $\mathcal{L}$, the vector $\mathbf{u}$ represents the discretization of the function u in space and b is a constant vector. We then decompose A into the form

$$A = L + D + U$$

where L, D and U are the strictly lower part, diagonal part and strictly upper part of A respectively. Defining

$$B = L + \frac{1}{2}D \text{ and } C = U + \frac{1}{2}D\,,$$

we can write the ADE scheme as the average of

$$(I - \Delta t\, B)\,\mathbf{p}^{n+1} = \mathbf{u}^n + \Delta t\, C\mathbf{u}^n + \Delta t\, b \tag{2}$$

and

$$(I - \Delta t\, C)\,\mathbf{q}^{n+1} = \mathbf{u}^n + \Delta t\, B\mathbf{u}^n + \Delta t\, b\,. \tag{3}$$

In equations (2) and (3), the matrices $(I - \Delta t B)$ and $(I - \Delta t C)$ are simply lower triangular and upper triangular, respectively. Therefore $\mathbf{p}^{n+1}$ and $\mathbf{q}^{n+1}$ can be updated fully explicitly using Gauss-Seidel iterations. Since the ADE method updates u^{n+1} by taking the average of solutions by (2) and (3), the updating formula of the ADE scheme in operator form can be written as

$$\begin{aligned}\mathbf{u}^{n+1} = \frac{1}{2}\left[(I - \Delta t B)^{-1}(I + \Delta t C) + (I - \Delta t C)^{-1}(I + \Delta t B)\right]\mathbf{u}^n \\ + \frac{1}{2}\left[(I - \Delta t B)^{-1} + (I - \Delta t C)^{-1}\right] b\,.\end{aligned} \tag{4}$$

Because of the triangular structure of the matrices B and C, the inverse of these operators can be easily determined using the forward substitution and backward substitution. The overall algorithm is shown in Algorithm 1 for completeness.

Even though such algorithm was first introduced only for the linear heat equation, the method is applicable to a much wider range of equations such as the first order nonlinear Hamilton-Jacobi equation, a fourth order nonlinear equation and curvature dependent flows [20]. The numerical scheme has various properties concerning the numerical stability and accuracy. We simply quote the following theorems from [20]. For the detailed technical analysis, we refer readers to the reference thereafter.

THEOREM 2.1 ([20]). *The ADE scheme has the following properties.*

1. *If the diagonal elements of A are all non-positive, the ADE scheme (4) is second order accurate in time.*
2. *If A is symmetric negative definite, the ADE scheme (4) is unconditionally stable.*
3. *If A is lower-triangular with all diagonal elements negative, the ADE scheme (4) is unconditionally stable.*

Algorithm 1: The ADE scheme for the general heat equation with the zero Dirichlet boundary condition.

Initialization: $n = 0$ and u^0;
while $n < N$ **do**
  $n \leftarrow n + 1$;
  Set $p_i^n = u_i^n$ and $q_i^n = u_i^n$ for $i = 0, 1, \cdots, M$ where $u_0^n = u_M^n = 0$;
  Set $p_0^{n+1} = p_M^{n+1} = q_0^{n+1} = q_M^{n+1} = 0$;
  For $i = 1, 2, \cdots, M-1$, solve

$$\frac{p_i^{n+1} - p_i^n}{\Delta t} = \frac{p_{i-1}^{n+1} - p_i^{n+1} - p_i^n + p_{i+1}^n}{\Delta x^2} + \tilde{b}_i^{n+\frac{1}{2}} .$$

  For $i = M-1, M-2, \cdots, 1$, solve

$$\frac{q_i^{n+1} - q_i^n}{\Delta t} = \frac{q_{i-1}^n - q_i^n - q_i^{n+1} + q_{i+1}^{n+1}}{\Delta x^2} + \tilde{b}_i^{n+\frac{1}{2}} .$$

  For $i = M-1, M-2, \cdots, 1$, compute $u_i^{n+1} = \frac{1}{2}\left(p_i^{n+1} + q_i^{n+1}\right)$.
end

2.2. The ADE scheme for the heat equation with a time dependent Dirichlet boundary condition. In the original derivation of the numerical scheme, one considers only the time-independent Dirichlet boundary condition such that the solution is fixed (to be zero) on the boundary at all time. In this subsection, we derive the ADE method for a general heat equation in one dimension given by

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + b(t) , \\ u(0,t) = f(t), u(l,t) = g(t) , \\ u(x,0) = v(x) . \end{cases} \tag{5}$$

In equation (5), both the source term and the boundary conditions are now functions of t. The corresponding derivation to higher dimensions is rather straightforward and will not be discussed in the current work. Let Ω be discretized using x_i's for $i = 0, 1, \cdots, M$ such that $x_i = i\Delta x$, $x_0 = 0$ and $x_M = l$. Using a time step Δt, we denote $t_n = n\Delta t$, for $n = 0, 1, \cdots, N$ such that $t_N = T$. Furthermore, we introduce the following notations given by

$$u_i^n = u(x_i, t_n) , \quad u_i^{n+\frac{1}{2}} = \frac{1}{2}(u_i^n + u_i^{n+1}) \text{ and } \tilde{u}_i^{n+\frac{1}{2}} = u\left(x_i, \frac{1}{2}(t_n + t^{n+1})\right) .$$

Then, equation (5) can be discretized using the ADE scheme approach given by

$$\begin{cases} \frac{u^{n+1} - u^n}{\Delta t} = ADE(\Delta_h, u^n, u^{n+1}) + \tilde{b}^{n+\frac{1}{2}} , \\ u_0^n = f^n, u_M^n = g^n , \\ u_i^0 = v_i , \ i = 0, 1, \cdots, M \text{ and } n = 0, 1, \cdots, N , \end{cases}$$

where the term $ADE(\Delta_h, u^n, u^{n+1})$ denotes the ADE treatment of the discrete Laplacian Δ_h involving both the directional sweepings and the averaging step which is

Algorithm 2: The ADE scheme for the general heat equation with the time-dependent Dirichlet boundary condition.

Initialization: $n = 0$ and u^0;
while $n < N$ **do**
  $n \leftarrow n + 1$;
  Set $p_i^n = u_i^n$ and $q_i^n = u_i^n$ for $i = 0, 1, \cdots, M$ where $u_0^n = f^n$ and $u_M^n = g^n$;
  Set $p_0^{n+1} = f^{n+1}$ and $q_M^{n+1} = g^{n+1}$ satisfying the boundary conditions at $x = x_0$ and x_M, respectively;
  For $i = 1, 2, \cdots, M-1$, solve

$$\frac{p_i^{n+1} - p_i^n}{\Delta t} = \frac{p_{i-1}^{n+1} - p_i^{n+1} - p_i^n + p_{i+1}^n}{\Delta x^2} + \tilde{b}_i^{n+\frac{1}{2}} . \tag{6}$$

  For $i = M-1, M-2, \cdots, 1$, solve

$$\frac{q_i^{n+1} - q_i^n}{\Delta t} = \frac{q_{i-1}^n - q_i^n - q_i^{n+1} + q_{i+1}^{n+1}}{\Delta x^2} + \tilde{b}_i^{n+\frac{1}{2}} . \tag{7}$$

  For $i = M-1, M-2, \cdots, 1$, compute

$$u_i^{n+1} = \frac{1}{2}\left(p_i^{n+1} + q_i^{n+1}\right) . \tag{8}$$

end

similar to that in the linear heat equation. To summarize, the method is given in Algorithm 2.

The scheme (6)-(8) is similar to the original ADE method for the heat equation with the time-independent Dirichlet boundary condition. The main difference is only in the treatment to those grid points adjacent to the boundary. In equation (6), p_i^n is updated in the ascending direction from $i = 1$ to $i = M-1$. For $i = 1$, since we have the boundary condition $u_0^{n+1} = f^{n+1}$, equation (6) can be written as

$$\frac{p_1^{n+1} - p_1^n}{\Delta t} = \frac{f^{n+1} - p_1^{n+1} - p_1^n + p_2^n}{\Delta x^2} + \tilde{b}_i^{n+\frac{1}{2}} . \tag{9}$$

Therefore, the quantity p_1^{n+1} is updated using the exact boundary condition f^{n+1} at $t = t_{n+1}$. For the grid point with the index $i = M-1$, we use the boundary condition $u_M^n = g^n$ and write equation (6) as

$$\frac{p_{M-1}^{n+1} - p_{M-1}^n}{\Delta t} = \frac{p_{M-2}^{n+1} - p_{M-1}^{n+1} - p_{M-1}^n + g^n}{\Delta x^2} + \tilde{b}_i^{n+\frac{1}{2}} . \tag{10}$$

Therefore, the quantity p_{M-1}^n is updated using the value of g at the previous time level given by g^n at $t = t_n$. Although we have the exact boundary condition at $t = t_{n+1}$, we use its value at $t = t_n$.

Similar consideration applies to the update of q according to equation (7) where we sweep through the index in a descending direction from $i = M-1$ to $i = 1$. For the index $i = M-1$, we have the boundary condition $u_M^{n+1} = g^{n+1}$ so that equation

(7) can be written as

$$\frac{q_{M-1}^{n+1} - q_{M-1}^{n}}{\Delta t} = \frac{q_{M-2}^{n} - q_{M-1}^{n} - q_{M-1}^{n+1} + g^{n+1}}{\Delta x^2} + \tilde{b}_i^{n+\frac{1}{2}} . \tag{11}$$

The quantity q_{M-1}^{n+1}, therefore, is updated using the exact boundary condition at the new time level $t = t_{n+1}$. For the left end point at the index $i = 1$, we impose the boundary condition $u_0^n = f^n$ for equation (7) and rewrite it as

$$\frac{q_1^{n+1} - q_1^{n}}{\Delta t} = \frac{f^{n} - q_1^{n} - q_1^{n+1} + q_2^{n+1}}{\Delta x^2} + \tilde{b}_i^{n+\frac{1}{2}} . \tag{12}$$

This implies that the quantity q_1^{n+1} is updated using the boundary condition imposed at the previous time level $t = t_n$.

Now we analyze the numerical accuracy of the proposed ADE scheme with the time-dependent boundary condition. One main property of this scheme is that the above updating formulas (9)-(12) at grid points adjacent to the boundary are designed to keep the symmetry of scheme (6)-(8). By Taylor expansion at $(x = i\Delta x, t = (n + \frac{1}{2})\Delta t)$, the leading order truncation error in p and q are given by

$$-\frac{\Delta t}{\Delta x} p_{tx} - \frac{1}{6}\Delta t \Delta x p_{txxx} - \frac{1}{24}\Delta t^2 \left(\frac{\Delta t}{\Delta x}\right) p_{tttx} + \Delta t^2 \left(\frac{1}{8} p_{ttxx} - \frac{1}{24} p_{ttt}\right) + \frac{1}{12}\Delta x^2 p_{xxxx}$$

and

$$\frac{\Delta t}{\Delta x} q_{tx} + \frac{1}{6}\Delta t \Delta x q_{txxx} + \frac{1}{24}\Delta t^2 \left(\frac{\Delta t}{\Delta x}\right) q_{tttx} + \Delta t^2 \left(\frac{1}{8} q_{ttxx} - \frac{1}{24} q_{ttt}\right) + \frac{1}{12}\Delta x^2 q_{xxxx} ,$$

respectively. After the averaging step (8), the leading error of the overall scheme (6)-(8) is therefore given by

$$\Delta t^2 \left(\frac{1}{8} u_{ttxx} - \frac{1}{24} u_{ttt}\right) + \frac{1}{12}\Delta x^2 u_{xxxx} .$$

In the above derivation, we assume that the function b can be evaluated at all half-time level given by $\tilde{b}_i^{n+\frac{1}{2}}$. In some applications, however, when we can only provide b at gridded levels $t = n\Delta t$ for some integer n, we approximate the quantity by $b_i^{n+\frac{1}{2}}$ which is second order in time. Then the leading error of the scheme (6)-(8) is therefore given by

$$\Delta t^2 \left(\frac{1}{16} u_{ttxx} - \frac{1}{24} u_{ttt} + \frac{1}{4} b_{tt}\right) + \frac{1}{12}\Delta x^2 u_{xxxx} ,$$

which is still $O(\Delta t^2, \Delta x^2)$ accurate.

Now, we discuss the overall computational complexity. Since the updating formulas involve only explicit operation at each individual grid point, the computational complexity of obtaining u_i^{n+1} for all i from u_i^n is of $O(M)$ with the total number of grid points is given by $M + 1$. This means that the complexity is in fact the same as typical explicit scheme for the heat equation. But because the method is unconditionally stable, it takes less time steps to reach the final time of the simulation.

2.3. The ADE scheme for the heat equation with the zero Neumann boundary condition. In this subsection, we discuss the implementation of the Neumann boundary condition. Even though we consider only the zero Neumann boundary condition, the algorithm can be easily modified to solve problems with non-zero Neumann boundary condition.

We consider the same problem given by equation (5) but replace the Dirichlet boundary condition by the zero Neumann conditions given by $u_x(0) = 0$ and $u_x(l) = 0$. We further assume that we have already obtained u^n and would like to use the ADE scheme to get u^{n+1}. Numerically, the zero Neumann boundary condition is typically approximated by standard finite difference and is given by $u_1^n = u_0^n$ and $u_M^n = u_{M-1}^n$ for all n. In the first sub-problem of the ADE scheme where we sweep through the solution in the ascending order, we need the boundary condition p_0^{n+1} at the updated time level t_{n+1}. We note that such value is actually an unknown value if we require the solution satisfies $p_0^{n+1} = p_1^{n+1}$. However, if we substitute this constraint to the updating formula of the ADE scheme for the first sub-problem, we can actually obtain

$$p_0^{n+1} = \left(1 - \frac{\Delta t}{\Delta x^2}\right) p_1^n + \frac{\Delta t}{\Delta x^2}\, p_2^n + \Delta t\, \tilde{b}_1^{n+\frac{1}{2}}\,. \tag{13}$$

Such choice of the boundary condition will automatically lead to the zero Neumann boundary condition at the left boundary. Similarly, for the second sub-problem when we sweep through the index in the descending order, we impose the following boundary condition at the right boundary given by

$$q_M^{n+1} = \frac{\Delta t}{\Delta x^2}\, q_{M-2}^n + \left(1 - \frac{\Delta t}{\Delta x^2}\right) q_{M-1}^n + \Delta t\, \tilde{b}_M^{n+\frac{1}{2}}\,. \tag{14}$$

Such choice will automatically lead to $q_M^{n+1} = q_{M-1}^{n+1}$. To summarize, we give the overall scheme in Algorithm 3. The scheme is extremely easy to implement. Comparing to Algorithm 2, we only need to modify boundary conditions by imposing $u_0^n = u_1^n$ and $u_M^n = u_{M-1}^n$, setting p_0^{n+1} using (13) and q_M^{n+1} using (14). The main part of the numerical implementation remains exactly the same.

3. Applications to partial differential equations with fractional derivatives. Because of the nice stability property of the ADE scheme, the method is especially suitable to problems for which standard finite difference methods require a restrictive time stability constraint. In this section, we discuss two applications of the ADE scheme related to fractional derivatives. The first one is a time-distributed order super-diffusive partial differential equation with the time dependent Dirichlet boundary condition. The second one is a reaction-diffusion system in a sub-diffusive regime with the zero Neumann boundary condition.

3.1. The time distributed order super-diffusive partial differential equation. Fractional differential equation (FDE) is a powerful technique in modeling an anomalous phenomena such as anomalous diffusion and anomalous transportation [28, 36, 29]. In the diffusion process, the mean squared displacement (MSD) can be written as an exponential function of the time t with a coefficient α. For the case where $\alpha = 1$, the diffusion process reduces back to the typical isotropic diffusion modeled by the heat equation $u_t = \Delta u$. The anomalous diffusion includes two types: sub-diffusion and super-diffusion. The diffusion process is called sub-diffusion (diffuse slower than normal) if $0 < \alpha < 1$, and is called super-diffusion (diffuse faster than

Algorithm 3: The ADE scheme for the general heat equation with the zero Neumann boundary condition.

Initialization: $n = 0$ and u^0;
while $n < N$ **do**
  $n \leftarrow n + 1$;
  Set $p_i^n = u_i^n$ and $q_i^n = u_i^n$ for $i = 0, 1, \cdots, M$ where $u_0^n = u_1^n$ and $u_M^n = u_{M-1}^n$;
  Set p_0^{n+1} using (13) and q_M^{n+1} using (14) ;
  For $i = 1, 2, \cdots, M-1$, solve

$$\frac{p_i^{n+1} - p_i^n}{\Delta t} = \frac{p_{i-1}^{n+1} - p_i^{n+1} - p_i^n + p_{i+1}^n}{\Delta x^2} + \tilde{b}_i^{n+\frac{1}{2}} .$$

  For $i = M-1, M-2, \cdots, 1$, solve

$$\frac{q_i^{n+1} - q_i^n}{\Delta t} = \frac{q_{i-1}^n - q_i^n - q_i^{n+1} + q_{i+1}^{n+1}}{\Delta x^2} + \tilde{b}_i^{n+\frac{1}{2}} .$$

  For $i = M-1, M-2, \cdots, 1$, compute $u_i^{n+1} = \frac{1}{2}\left(p_i^{n+1} + q_i^{n+1}\right)$.
end

normal) if $1 < \alpha < 2$. Mathematically, these anomalous diffusion processes are modeled using a time fractional order differential equation [6]. More recently, some efforts are spent to solve more complicated problems where the diffusion process contains diffusions in various time scales so that the diffusion speed is modeled by a range of parameter α. This leads to the so-called time distributed order differential equations [8]. In this section, we consider the following time distributed-order super-diffusive problem given by

$$\begin{cases} \int_1^2 w(\gamma) {}_0^C D_t^\gamma u(x,y,t) d\gamma = \Delta u(x,y,t) + F(x,y,t), (x,y) \in \Omega, 0 < t \le T, \\ u(x,y,0) = 0, u_t(x,y,0) = 0, (x,y) \in \Omega\,, \\ u(x,y,t) = \psi(x,y,t), (x,y) \text{ on } \partial\Omega, 0 \le t \le T\,, \end{cases} \tag{15}$$

with $\Omega = \{(x,y)|0 < x < L_1, 0 < y < L_2\}$, and $\partial\Omega$ being the boundary of Ω. The functions $F(x,y,t)$ and $\psi(x,y,t)$ are given to model the time-dependent source term and the initial condition, respectively. The function $w(\gamma) \ge 0$ with $\int_1^2 w(\gamma) d\gamma = c_0 > 0$ determines the weights among various diffusion speed scales. In equation (15), the derivative ${}_0^C D_t^\gamma u(x,y,t)$ is the γ-th order time Caputo fractional derivative defined as

$${}_0^C D_t^\gamma u(x,y,t) = \begin{cases} u_t(x,y,t) - u_t(x,y,0), \text{ if } \gamma = 1\,, \\ \frac{1}{\Gamma(2-\gamma)} \int_0^t (t-\xi)^{1-\gamma} \frac{\partial^2 u}{\partial \xi^2}(x,y,\xi) d\xi, 1 < \gamma < 2, \\ u_{tt}(x,y,t), \text{ if } \gamma = 2\,. \end{cases} \tag{16}$$

Consider a rectangle domain $\Omega = \{(x,y)|0 \le x \le L_1, 0 \le y \le L_2\}$. Let Δx and Δy be the mesh size in the x- and the y-direction, respectively. Define $x_i = i\Delta x$ and $y_j = j\Delta y$ for $0 \le i \le M_1$ and $0 \le j \le M_2$ such that $x_{M_1} = L_1$ and $y_{M_2} = L_2$. Denote the set $\omega = \{(i,j)|1 \le i \le M_1 - 1, 1 \le j \le M_2 - 1\}$ and $\omega_0 = \{(i,j)|(x_i, y_j) \in \partial\Omega\}$ and $\omega_h = \omega \cap \omega_0$. For any $v \in \Omega_h$, the discretization of Ω, we introduce the following

difference notations:

$$\delta_x v_{i-\frac{1}{2}j} = \frac{1}{\Delta x}(v_{ij} - v_{i-1j}),\ \delta_x^2 v_{ij} = \frac{1}{\Delta x}\left(\delta_x v_{i+\frac{1}{2}j} - \delta_x v_{i-\frac{1}{2}j}\right),$$
$$\delta_y v_{i-\frac{1}{2}j} = \frac{1}{\Delta y}(v_{ij} - v_{ij-1}),\ \delta_y^2 v_{ij} = \frac{1}{\Delta y}\left(\delta_x v_{ij+\frac{1}{2}} - \delta_y v_{ij-\frac{1}{2}}\right),$$
$$\Delta_h v_{ij} = \delta_x^2 v_{ij} + \delta_y^2 v_{ij}\,.$$

In the time space, we use Δt to denote our time step. Define $t_n = n\Delta t, n = 0, 1, \cdots, N$ such that $t_N = T$. For the integral on the fractional derivative in time, we divide the integration domain $[1, 2]$ into J subintervals with length $\Delta\gamma = \frac{1}{J}$ and define $\gamma_l = 1 + l\Delta\gamma, l = 0, 1, \cdots, J$. Then the integration will be approximated using Trapezoidal rule:

$$\int_1^2 s(\gamma)d\gamma = \Delta\gamma \sum_{l=0}^{J} c_l s_{\gamma_l} + O(\Delta\gamma^2)$$

with $c_l = \frac{1}{2}$ for $l = 0$ or J, and $c_l = 1$ otherwise.

Following a similar strategy as in [8], for $\alpha \in [0, 1]$, we define $g_k^{(\alpha)}$, $\lambda_k^{(\alpha)}$ and μ as

$$g_0^{(0)} = 1,\ g_k^{(0)} = 0\,,\ g_0^{\alpha} = 1,\ g_k^{\alpha} = \left(1 - \frac{\alpha+1}{k}\right) g_{k-1}^{(\alpha)}\,,\ \lambda_0^{(\lambda)} = \left(1 + \frac{\alpha}{2}\right) g_0^{(\alpha)}\,,$$
$$\lambda_k^{(\alpha)} = \left(1 + \frac{\alpha}{2}\right) g_k^{(\alpha)} - \frac{\alpha}{2} g_{k-1}^{(\alpha)}\,,\ \mu = \Delta\gamma \sum_{l=0}^{J} c_l w(\gamma_l)\tau^{-\alpha_l}\lambda_0^{(\alpha_l)}\,,$$

for $k \geq 1$ and $0 < \alpha \leq 1$. Denote $\alpha_l = \gamma_l - 1$, a simple finite difference scheme applied to equation (15) is given by

$$\begin{cases} \Delta\gamma \sum_{l=0}^{J} c_l w(\gamma_l)\Delta t^{-\alpha_l} \sum_{k=0}^{n} \lambda_k^{(\alpha_l)} \delta_t u_{ij}^{n-k+\frac{1}{2}} = \Delta_h u_{ij}^{n+\frac{1}{2}} + F_{ij}^{n+\frac{1}{2}}, (i,j) \in \omega, 1 \leq n \leq N, \\ u_{ij}^0 = 0, (i,j) \in \omega \ \text{ and } \ u_{ij}^n = \psi(x_i, y_j, t_n), (i,j) \in \omega_0, 0 \leq n \leq N-1\,, \end{cases}$$

where $\Delta_h u_{ij}^{n+\frac{1}{2}}$ is an approximation of the Laplacian of the solution at the half time level $t_{n+\frac{1}{2}}$ using methods such as central difference. The main issue, however, is that it is not easy to find an efficient solver to invert for u^{n+1}. In particular, ADI scheme is not directly applicable. In a recent work, [8] has introduced an extra regularization to the discretization by $\frac{\Delta t^2}{4\mu}\delta_x^2\delta_y^2\delta_t u_{ij}^{n+\frac{1}{2}}$ which also nicely coupled with the ADI scheme,

$$\begin{cases} \Delta\gamma \sum_{l=0}^{J} c_l w(\gamma_l)\Delta t^{-\alpha_l} \sum_{k=0}^{n} \lambda_k^{(\alpha_l)} \delta_t u_{ij}^{n-k+\frac{1}{2}} + \frac{\Delta t^2}{4\mu}\delta_x^2\delta_y^2\delta_t u_{ij}^{n+\frac{1}{2}} = \Delta_h u_{ij}^{n+\frac{1}{2}} + F_{ij}^{n+\frac{1}{2}}\,, \\ u_{ij}^0 = 0, (i,j) \in \omega \ \text{ and } \ u_{ij}^n = \psi(x_i, y_j, t_n), (i,j) \in \omega_0, 0 \leq n \leq N-1\,. \end{cases} \tag{17}$$

This ADI scheme (17) as developed in [8] has been proven to give an approximation to (15) with the error $O(\Delta t^2 + \Delta x^2 + \Delta y^2 + \Delta\gamma^2)$. In the derivation of (17), the extra regularization term $\frac{\Delta t^2}{4\mu}\delta_x^2\delta_y^2\delta_t u_{ij}^{n+\frac{1}{2}}$ is manually incorporated into the equation. Such term is necessary in the ADI scheme solely originated from the construction of the numerical method but is unnecessary in our ADE approach.

Our proposed ADE finite difference scheme is given by

$$\begin{cases} \Delta\gamma \sum_{l=0}^{J} c_l w(\gamma_l) \Delta t^{-\alpha_l} \sum_{k=0}^{n} \lambda_k^{(\alpha_l)} \delta_t u_{ij}^{n-k+\frac{1}{2}} = ADE(\Delta_h, u^n, u^{n+1}) + F_{ij}^{n+\frac{1}{2}}, \\ u_{ij}^0 = 0, (i,j) \in \omega \text{ and } u_{ij}^n = \psi(x_i, y_j, t_n), (i,j) \in \omega_0, 0 \le n \le N-1. \end{cases} \tag{18}$$

Note that this expression can be reorganized as

$$\begin{cases} \mu \delta_t u_{ij}^{n+\frac{1}{2}} = ADE(\Delta_h, u^n, u^{n+1}) \\ \qquad\qquad + \Delta\gamma \sum_{l=0}^{J} c_l w(\gamma_l) \Delta t^{-\alpha_l} \sum_{k=1}^{n} \lambda_k^{(\alpha_l)} \delta_t u_{ij}^{n-k+\frac{1}{2}} + F_{ij}^{n+\frac{1}{2}}, \\ u_{ij}^0 = 0, (i,j) \in \omega \text{ and } u_{ij}^n = \psi(x_i, y_j, t_n), (i,j) \in \omega_0, 0 \le n \le N-1 \end{cases} \tag{19}$$

which is a diffusion equation of $u^{n+\frac{1}{2}}$. Assume that we are given u^k, for $k = 0, 1, \cdots, n$ and want to compute u^{n+1}. In equation (19), the time derivative term $\delta_t u_{ij}^{k+\frac{1}{2}}$ can be approximated by

$$\delta_t u_{ij}^{k+\frac{1}{2}} = \frac{u_{ij}^{k+1} - u_{ij}^k}{\Delta t}, \tag{20}$$

for $k = 0, 1, \cdots, n$, with the second order accuracy in time. We can analytically determine $F_{ij}^{n+\frac{1}{2}}$ since the expression of F is given. Substituting the approximation (20) to the equation (19) and applying the ADE method, we finally get the following Algorithm 4.

To end this section, we present two theorems of the stability and convergence of (18). The proofs are similar to those as shown in [8]. We recommend interested readers to the reference and thereafter.

THEOREM 3.1. *Let $\{u_{ij}^{n+1} | (i,j) \in \omega_h, 0 \le N\}$ be the solution of the following difference scheme*

$$\begin{cases} \Delta\gamma \sum_{l=0}^{J} c_l w(\gamma_l) \Delta t^{-\alpha_l} \sum_{k=0}^{n} \lambda_k^{(\alpha_l)} \delta_t u_{ij}^{n-k+\frac{1}{2}} = ADE(\Delta_h, u^n, u^{n+1}) + G_{ij}^{n+1}, \\ u_{ij}^0 = \phi_{i,j}, (i,j) \in \omega \text{ and } u_{ij}^n = 0, (i,j) \in \omega_0, 0 \le n \le N-1, \end{cases}$$

then it holds

$$\|\Delta_h u^{n+1}\|^2 \le \exp(T) \left[3\|\Delta_h u^0\|^2 + 4(\|G^1\|^2 + \max_{1 \le k \le n+1} \|G^k\|^2) + 4\Delta \sum_{k=1}^{n} \|\delta_t G^{k+\frac{1}{2}}\|^2 \right]$$

for $1 \le n \le N-1$ where

$$\|G^n\|^2 = \Delta x \Delta y \sum_{i=1}^{M_1-1} \sum_{j=1}^{M_2-1} (G_{ij}^n)^2 \text{ and } \|\delta_t G^{n+\frac{1}{2}}\|^2 = \Delta x \Delta y \sum_{i=1}^{M_1-1} \sum_{j=1}^{M_2-1} (\delta_t G_{ij}^{n+\frac{1}{2}})^2.$$

THEOREM 3.2. *Let $u(x,y,t)$ be the C^2-solution of the problem (15) and $\{u_{ij}^n | (i,j) \in \omega_h, 0 \le n \le N\}$ be the solution of (18). Denote $e_{ij}^n = U_{ij}^n - u_{ij}^n, (i,j) \in \omega_h, 0 \le n \le N$. Then we have*

$$\|e^n\|_\infty \le 2C \cdot \exp(T/2) \sqrt{(2+T)L_1 L_2} (\Delta t^2 + \Delta x^2 + \Delta y^2 + \Delta\gamma^2),$$

Algorithm 4: The ADE scheme for the two dimensional time distributed order super-diffusive PDE.

Initialization: $n = 0$ and u^0;
while $n < N$ **do**
$n \leftarrow n + 1$;
Given u^k for $k = 0, 1, \cdots, n$ and also the function F;
Compute $\delta_t u^{n-k+\frac{1}{2}}, k = 1, 2, \cdots, n$ and $F^{n+\frac{1}{2}}$;
For $i = 1, 2, \cdots, M_1 - 1$ and $j = 1, 2, \cdots, M_2 - 1$, compute

$$\tilde{b}_{ij}^{n+\frac{1}{2}} = -\Delta\gamma \sum_{l=0}^{J} c_l w(\gamma_l) \Delta t^{-\alpha_l} \sum_{k=1}^{n} \lambda_k^{(\alpha_l)} \delta_t u_{ij}^{n-k+\frac{1}{2}} + F_{ij}^{n+\frac{1}{2}} .$$

Set $p_{ij}^n = q_{ij}^n = v_{ij}^n = w_{ij}^n = u_{ij}^n$, for $i = 1, 2, \cdots, M_1 - 1$ and $j = 1, 2, \cdots, M_2 - 1$;
Set $p_{ij}^k = \psi_{ij}^k, q_{ij}^k = \psi_{ij}^k, v_{ij}^k = \psi_{ij}^k, w_{ij}^k = \psi_{ij}^k, \forall (i, j) \in \omega_0, k = n + 1, n$;
For $i = 1, 2, \cdots, M_1 - 1,\ j = 1, 2, \cdots, M_2 - 1$, solve

$$\frac{p_{ij}^{n+1} - p_{ij}^n}{\Delta t} = \frac{p_{i-1j}^{n+1} - p_{ij}^{n+1} - p_{ij}^n + p_{i+1j}^n}{\Delta x^2} + \frac{p_{ij-1}^{n+1} - p_{ij}^{n+1} - p_{ij}^n + p_{ij+1}^n}{\Delta y^2} + \tilde{b}_i^{n+\frac{1}{2}} .$$

For $i = 1, 2, \cdots, M_1 - 1,\ j = M_2 - 1, M_2 - 2, \cdots, 1$, solve

$$\frac{q_{ij}^{n+1} - q_{ij}^n}{\Delta t} = \frac{q_{i-1j}^{n+1} - q_{ij}^{n+1} - q_{ij}^n + q_{i+1j}^n}{\Delta x^2} + \frac{q_{ij-1}^n - q_{ij}^n - q_{ij}^{n+1} + q_{ij+1}^{n+1}}{\Delta y^2} + \tilde{b}_i^{n+\frac{1}{2}} .$$

For $i = M_1 - 1, M_1 - 2, \cdots, 1,\ j = 1, 2, \cdots, M_2 - 1$, solve

$$\frac{v_{ij}^{n+1} - v_{ij}^n}{\Delta t} = \frac{v_{i-1j}^n - v_{ij}^n - v_{ij}^{n+1} + v_{i+1j}^{n+1}}{\Delta x^2} + \frac{v_{ij-1}^{n+1} - v_{ij}^{n+1} - v_{ij}^n + v_{ij+1}^n}{\Delta y^2} + \tilde{b}_i^{n+\frac{1}{2}} .$$

For $i = M_1 - 1, M_1 - 2, \cdots, 1,\ j = M_2 - 1, M_2 - 2, \cdots, 1$, solve

$$\frac{w_{ij}^{n+1} - w_{ij}^n}{\Delta t} = \frac{w_{i-1j}^n - w_{ij}^n - w_{ij}^{n+1} + w_{i+1j}^{n+1}}{\Delta x^2} + \frac{w_{ij-1}^n - w_{ij}^n - w_{ij}^{n+1} + w_{ij+1}^{n+1}}{\Delta y^2} + \tilde{b}_i^{n+\frac{1}{2}} .$$

Compute $u_{ij}^{n+1} = \frac{1}{4}\left(p_{ij}^{n+1} + q_{ij}^{n+1} + v_{ij}^{n+1} + w_{ij}^{n+1}\right)$ for $(i, j) \in \omega$.
end

for $1 \le n \le N$ where C is some constant.

Indeed, numerical methods for solving fractional derivative equations grow tremendously in recent years. For example, see [7, 4, 18, 24, 33, 23, 22, 3]. We are not aiming to provide a complete review of the numerical approaches to the equation, nor to compare the efficiency or the accuracy of these numerical schemes. The main purpose of this section is to show that it is straightforward to implement the ADE scheme to solve this class of equations and to demonstrates the effectiveness of the scheme. We leave it as a future work to investigate the performance among

various numerical schemes.

3.2. A reaction-diffusion system in a sub-diffusive regime. In the second application, we apply the ADE scheme to solve a sub-diffusive reaction-diffusion system for pattern formation based on diffusion-driven instability. We consider the Turing pattern [34] which has been a very successive model to explain the patterns on various surfaces [16, 30], porous media flows [5], and biological systems [27]. One famous class of such reaction-diffusion systems is the so-called two-component activator-inhibitor system. In this system under some conditions, one component stimulates the growth rate of both components while the other one inhibits their productions. In a recent work, [15] has generalized the standard activator-inhibitor system developed in [2] by replacing the typical diffusion by sub-diffusion. The system reads as

$$\begin{cases} \frac{\partial^\alpha u}{\partial t^\alpha} = D\delta\Delta u + a_{11}u + a_{12}v - r_2uv - (a_{11}r_1)uv^2\,, \\ \frac{\partial^\beta v}{\partial t^\beta} = \delta\Delta v + a_{21}u + a_{22}v + r_2uv + (a_{11}r_1)uv^2\,, \end{cases} \tag{21}$$

for $0 < \alpha, \beta \leq 1$. The parameters D and δ are two coefficients governing the diffusion associated to two components. The derivative $\frac{\partial^\alpha}{\partial t^\alpha}$ is the Caputo-type fractional derivative of order α. For problems with finite domain, we impose the zero Neumann boundary condition on the boundary of the computational domain.

To approximate the fractional derivative numerically, we follow the idea as introduced in [14] by

$$\left.\frac{\partial^\alpha u_{ij}}{\partial t^\alpha}\right|_{t_{n+1}} = \sum_{m=0}^{n+1}(-1)^m\binom{\alpha}{m}\frac{u_{ij}(t_{n+1-m}) - u_{ij}(t_0)}{(\Delta t)^\alpha}\,. \tag{22}$$

In our current implementation, we treat the terms related to v in the first equation in (21) explicitly when updating the quantity u, while we take all u-related terms in the second equation in (21) explicitly when updating v. Then equation (21) can be discretized as

$$\begin{cases} \dfrac{u_{ij}^{n+1} - u_{ij}^0}{\Delta t^\alpha} = D\delta\, ADE(\Delta_h, u^n, u^{n+1}) + a_{11}\dfrac{u_{ij}^{n+1} + u_{ij}^n}{2} + a_{12}v_{ij}^n - r_2u_{ij}^nv_{ij}^n \\ \qquad\qquad -(a_{11}r_1)u_{ij}^n(v_{ij}^n)^2 - \displaystyle\sum_{m=1}^{n}(-1)^m\binom{\alpha}{m}\frac{u_{ij}^{n+1-m} - u_{ij}^0}{(\Delta t)^\alpha}, \\ \dfrac{v_{ij}^{n+1} - v_{ij}^0}{\Delta t^\beta} = \delta\, ADE(\Delta_h, v^n, v^{n+1}) + a_{21}u^n + a_{22}\dfrac{v_{ij}^{n+1} + v_{ij}^n}{2} + r_2u_{ij}^nv_{ij}^n \\ \qquad\qquad +(a_{11}r_1)u_{ij}^n(v_{ij}^n)^2 - \displaystyle\sum_{m=1}^{n}(-1)^m\binom{\beta}{m}\frac{v_{ij}^{n+1-m} - v_{ij}^0}{(\Delta t)^\beta}\,. \end{cases}$$

Indeed, various other ADE forms are possible for this equation in treating the nonlinear terms. But we tend to concentrate only on the effectiveness of the simple ADE discretization to complicated equation. More detailed studies on the ADE discretization of a general nonlinear system of PDEs will be given in the future.

4. Numerical experiments. In this section, we test the accuracy of the ADE algorithm. For one dimensional examples, we use $\Delta x = 1/M$ such that there are $M + 1$ grid points. For two dimensional examples, we consider a square domain and use $M + 1$ grid points in each dimension to discretize the domain. We use T to represent the final time and N is the number of time levels after the initial state (i.e.

$\Delta t = T/N$). The computed solution and the exact solution at $t = T$ will be denoted by u_T and u_T^*, respectively. The L_2 and L_∞ norms are computed at $t = T$ defined by $\|u_T - u_T^*\|_2$ and $\|u_T - u_T^*\|_\infty$.

4.1. Heat equations with time-dependent Dirichlet boundary condition. In our first test, we solve the one dimensional heat equation (5). We consider the source $F = \cos(x + t) - \sin(x + t)$ with the initial condition $v = \cos(x)$. The computational domain is $\Omega = [-\pi, \pi]$. The boundary conditions are $f(t) = \cos(t - \pi)$ and $g(t) = \cos(t + \pi)$ so that the exact solution is given as $u = \cos(x + t)$. First we fix our grid size, using $M = 100$, to check the accuracy order with respect to Δt. We set $T = 2$. The errors and accuracy order for different N are shown in Table 1. Then we fix the time step by using $N = 10^5$ and vary Δx to test the accuracy in space. The final time T is set to be 0.5. The errors and accuracy orders are shown in Table 2. We see from these tables that the accuracy in both time and space are approximately second order defined using both the L_2 norm and L_∞ norm.

N	$\|u_T - u_T^*\|_2$	rate	$\|u_T - u_T^*\|_\infty$	rate
20	1.46×10^{-1}		4.18×10^{-1}	
40	4.08×10^{-2}	1.84	$1.18\ \times10^{-1}$	1.82
80	1.05×10^{-2}	1.96	3.05×10^{-2}	1.95
160	2.65×10^{-3}	1.99	$7.72\ \times10^{-3}$	1.98

TABLE 1

(Example 4.1, one dimensional case) To test the convergence in time, we take $T = 2$ and $M = 100$. The table shows the L_2- and L_∞-errors using different N.

M	$\|u_T - u_T^*\|_2$	rate	$\|u_T - u_T^*\|_\infty$	rate
10	1.85×10^{-2}		1.57×10^{-2}	
20	2.88×10^{-3}	2.68	$3.53\ \times10^{-3}$	2.15
40	4.78×10^{-4}	2.59	8.42×10^{-4}	2.07
80	8.19×10^{-5}	2.55	$2.05\ \times10^{-4}$	2.04

TABLE 2

(Example 4.1, one dimensional case) To test the convergence in space, we take $T = 0.5$ and $N = 10^5$. The table shows the L_2- and L_∞-errors using different M.

N	$\|u_T - u_T^*\|_2$	rate	$\|u_T - u_T^*\|_\infty$	rate
40	1.12×10^{-1}		7.13×10^{-2}	
80	3.77×10^{-2}	1.57	2.38×10^{-2}	1.58
160	1.05×10^{-2}	1.84	$6.57\ \times10^{-3}$	1.86
320	2.59×10^{-3}	2.02	$1.63\ \times10^{-3}$	2.01

TABLE 3

(Example 4.1, two dimensional case) To test the convergence in time, we take $T = 1$ and $M = 50$. The table shows the L_2- and L_∞-errors using different N.

Now we turn to a higher dimensional case where we solve the two dimensional heat equation with the source term given by $b(x, y, t) = 2(t + t^2)\sin(x + y)$, the initial condition $r(x, y) = 0$ and the time dependent boundary condition $h(x, y, t) = t^2 \sin(x + y)$. The exact solution is given by $u = t^2 \sin(x + y)$. We set the final time $T = 1$. First we fix $M = 50$ to test the accuracy order in time. The L_2 and L_∞ errors

M	$\|u_T - u_T^*\|_2$	rate	$\|u_T - u_T^*\|_\infty$	rate
10	5.20×10^{-3}		3.09×10^{-3}	
20	1.18×10^{-3}	2.14	$7.22\ \times10^{-4}$	2.10
40	2.81×10^{-4}	2.07	1.73×10^{-4}	2.06
80	6.83×10^{-5}	2.04	$4.21\ \times10^{-5}$	2.04

TABLE 4

(Example 4.1, two dimensional case) To test the convergence in space, we take $T = 1$ and $N = 10^5$. The table shows the L_2- and L_∞-errors using different M.

are shown in Table 3. We can see that the method is almost second order accurate in time. Then we fix $N = 10^5$ and test the convergence in space. These results are shown in Table 4. We observe also that the method gives approximately second order accurate solution in space.

N	$\|u_T - u_T^*\|_2$	rate	$\|u_T - u_T^*\|_\infty$	rate
500	3.22×10^{-2}		1.42×10^{0}	
1000	1.73×10^{-2}	0.90	$7.62\ \times10^{-1}$	0.90
2000	5.29×10^{-3}	1.71	2.34×10^{-1}	1.70
4000	1.42×10^{-3}	1.90	$6.36\ \times10^{-2}$	1.88

TABLE 5

(Example 4.2) To test the convergence in time, we take $T = 2$ and $M = 1000$. The table shows the L_2- and L_∞-errors using different N.

M	$\|u_T - u_T^*\|_2$	rate	$\|u_T - u_T^*\|_\infty$	rate
10	1.14×10^{-1}		5.13×10^{-1}	
20	3.48×10^{-2}	1.71	$2.46\ \times10^{-1}$	1.06
40	1.15×10^{-2}	1.60	1.21×10^{-1}	1.02
80	3.92×10^{-3}	1.55	$5.97\ \times10^{-2}$	1.02

TABLE 6

(Example 4.2) To test the convergence in space, we take $T = 2$ and $N = 10^5$. The table shows the L_2- and L_∞-errors using different M.

4.2. The one dimensional heat equation with the Neumann boundary condition. In this example, we apply the ADE scheme to a problem with the Neumann boundary condition. Consider the following problem

$$\begin{cases} u_t = \Delta u + \cos(\pi x) + \pi^2 t \cos(\pi x)\,, \\ u(x,0) = 0 \ \text{ and } \ u_x(0) = u_x(1) = 0\,. \end{cases}$$

The exact solution to this problem is given by $u = t\cos(\pi x)$. We consider the computational domain $\Omega = [0,1]$ and the final time $T = 2$. First, we fix $M = 1000$ and test the convergence in time using different time steps and N. The error in the numerical solutions are shown in Table 5. We also fix $N = 10^5$ and vary M to see the convergence in the spatial direction. The results are shown in Table 6.

4.3. The two dimensional time-distributed super-diffusive equation with the time-dependent Dirichlet boundary condition. In this example, we solve the time-distributed super-diffusive equation (15) using the formulation (19) by

N	$\|u_T - u_T^*\|_2$	rate	$\|u_T - u_T^*\|_\infty$	rate
10	5.60×10^{-2}		2.58×10^{-2}	
20	1.66×10^{-2}	1.76	7.65×10^{-3}	1.76
40	4.50×10^{-3}	1.88	2.08×10^{-3}	1.88
80	1.17×10^{-3}	1.94	5.40×10^{-4}	1.95

TABLE 7

(Example 4.3) To test the convergence in time, we take $T = 0.5$, $M = 100$ and $J = 200$. The table shows the L_2- and L_∞-errors using different N.

the ADE method. We consider the computation domain $\Omega = (0, \pi)^2$. The functions are chosen as

$$w(\gamma) = \Gamma(7-\gamma)\,,\ F(x,y,t) = 128t^4 \sin(x+y)\left[\frac{360(t-1)}{\ln t} + t^2\right],$$
$$\psi(x,y,t) = 64t^6 \sin(x+y)$$

so that the exact solution to the problem can be analytically found and is given by $u^*(x,y,t) = \psi(x,y,t)$. First we fix $M = 100$ and $J = 200$ to test the accuracy order in time. We set $T = 0.5$. The result is shown in Table 7. The L_2 and L_∞ norm are both almost second order.

4.4. The two dimensional sub-diffusive activator-inhibitor system with the zero Neumann boundary condition. In this example, we use the ADE scheme to solve the sub-diffusive activator-inhibitor system using 100×100 grid points with $\Delta x = 1$. The coefficient is chosen as $a_{11} = 0.899$, $a_{22} = -0.91$, $a_{12} = 1$, $a_{21} = -a_{11}$, $\delta = 2$, $D = 0.516$, $\alpha = 0.92$ and $\beta = 0.88$. With this setting, there is only one fixed point for this system given by $u = v = 0$. For the initial condition, we add some uniformly distributed noise ranging from -0.1 to 0.1. In equation (22) when approximating the fractional derivative, we find that the number of terms in the expression grows rapidly in time yet the magnitude of some of these terms are rather small. To improve the computational efficiency, we follow the approach in [15] and keep only the latest 800 terms whose binomial coefficient $\binom{\alpha}{m}$ or $\binom{\beta}{m}$ is larger than 10^{-7}. In our implementation, we use the time step $\Delta t = 0.08$. Our result is shown in Figure 1. The method as stated in [15] requires approximately 3.8×10^5 iterations to reach the steady state, while our ADE scheme takes only 2×10^4 iterations (roughly 1/20 of the former method) to obtain the final pattern since the unconditionally stable property in the ADE method allows us to use a significantly larger time step in updating the solution.

5. Conclusion. The ADE scheme is an unconditionally stable fully explicit AOS scheme and can efficiently solve time-evolution equations. We have extended the ADE scheme to time evolution equations with the time dependent Dirichlet boundary condition and the Neumann boundary condition. We also present various numerical examples including the heat equation with the time dependent Dirichlet boundary condition and the Neumann boundary condition, and two applications involving fractional derivatives including the time-distributed order super-diffusive equation and a reaction-diffusion system in the sub-diffusive regime. These numerical tests have demonstrated that the numerical approach is simple to implement and is computational efficient for a wide range of applications.

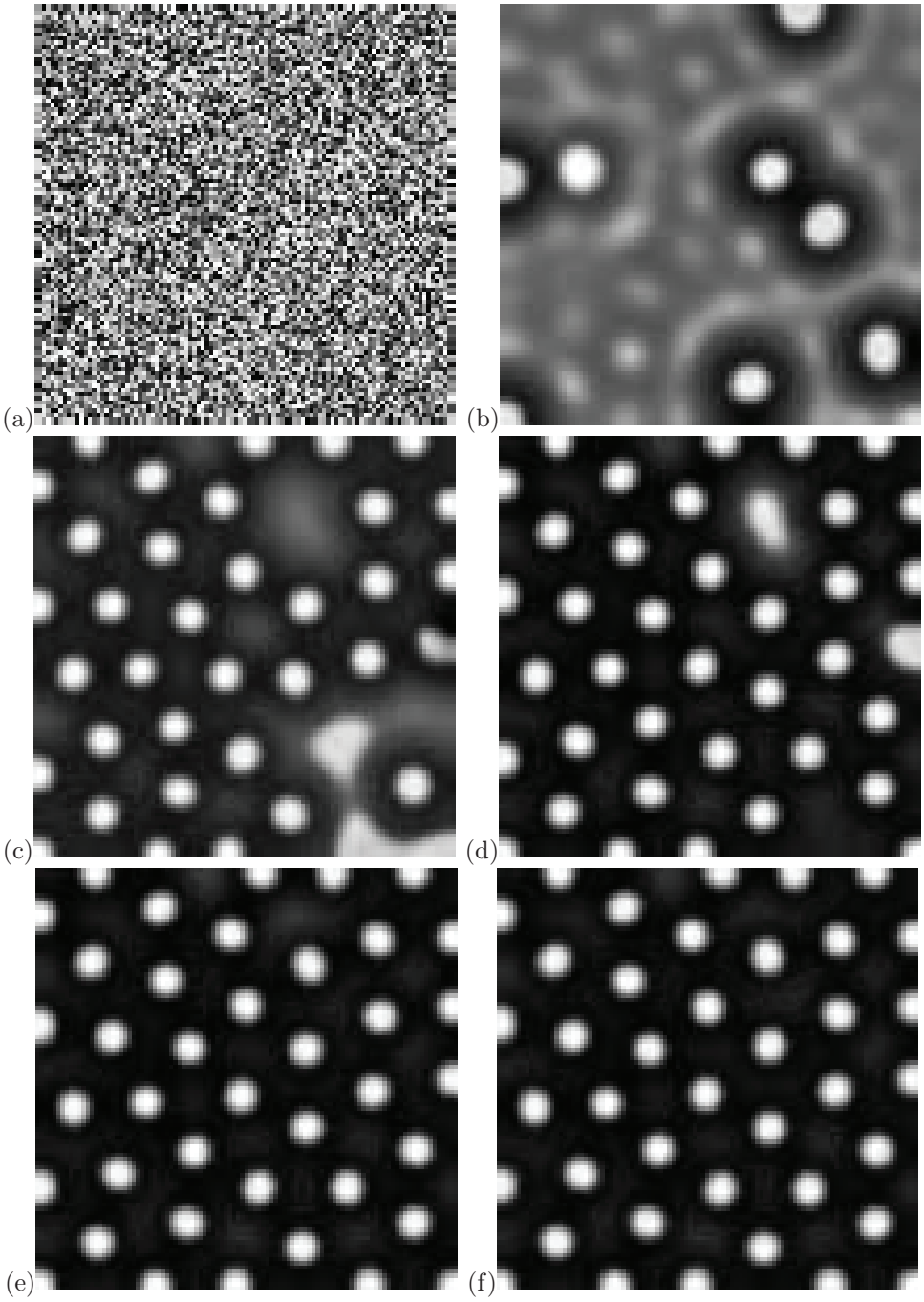

FIG. 1. *(Example 4.4) The initial condition is shown in (a). Turing pattern obtained after (b) 2000 iterations, (c) 6000 iterations, (d) 12000 iterations, (e) 16000 iterations and (f) 20000 iterations.*

Acknowledgment. The work of Leung was supported in part by the Hong Kong RGC grants 16303114 and 16309316.

REFERENCES

[1] H. Z. Barakat and J. A. Clark, *On the solution of the diffusion equations by numerical methods*, Journal of Heat transfer, 88:4 (1996), pp. 421–427.

[2] R. A. Barrio, C. Varea, and J. L. Aragón, *A two-dimensional numerical study of spatial pattern formation in interacting turing systems*, Bulletin of mathematical biology, 61:3(1999), pp. 483–505.

[3] O. Defterli, M. D'Elia, Q. Du, M. Gunzburger, R. Lehoucq, and M. M. Meerschaert, *Fractionl diffusion on bounded domains*, Fractional Calculus and Applied Analysis, 18:2 (2015), pp. 342–360.

[4] K. Diethelm, N. J. Ford, A. D. Freed, and Y. Luchko, *Algorithms for the fractional calculus: a selection of numerical methods*, Computer methods in ap plied mechanics and engineering, 194(6-8) (2005), pp. 743–773.

[5] G. Drazer and D. H. Zanette, *Experimental evidence of power-law trapping-time distributions in porous media*, Physical Review E, 60:5 (1999), 5858.

[6] C. H. Eab and S. C. Lim, *Fractional langevin equations of distributed order*, Physical Review E, 83:3 (2011), 031136.

[7] N. J. Ford and A. C. Simpson, *The numerical solution of fractional differential equations: speed versis accuracy*, Numer. Algorithms, 26:4 (2001), pp. 333–346.

[8] G. Gao and Z. Sun, *Two alternating direction implicit difference schemes for solving the two-dimensional time distributed-order wave equations*, Journal of Scientific Computing, 69:2 (2016), pp. 506–531.

[9] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, SIAM, Pennsylvania, US., 1989.

[10] R. Glowinski, S. Leung, and J. Qian, *A penalization-regularization-operator splitting method for eikonal based traveltime tomography*, SIAM J. Imaging Sciences, 8:2 (2015), pp. 1263–1292.

[11] R. Glowinski, S. Leung, and J. Qian, *Operator-splitting based fast sweeping methods for isotropic wave propagation in a moving fluid*, SIAM J. Sci. Comput., 38:2 (2016), pp. A1195–A1223.

[12] R. Glowinski, S. Leung, and J. Qian, *A simple explicit operator-splitting method for effective hamiltonians*, SIAM J. Sc. Comput., 2017.

[13] R. Glowinski, S. Osher, and W. Yin, *Splitting Methods in Communication, Imaging, Science, and Engineering Splitting methods in communication, imaging, science, and engineering*, Springer, 2016.

[14] R. Gorenflo and E. A. Abdel-Rehim. Convergence of the grünwald–letnikov scheme for time-fractional diffusion. *Journal of computational and applied mathematics*, 205(2):871–881, 2007.

[15] D. Hernández, C. Varea, and R. A. Barrio, *Dynamics of reaction-diffusion systems in a subdiffusive regime*, Physical Review E, 79:2 (2009), 026109.

[16] S. Kondo and R. Asai, *A reaction–diffusion wave on the skin of the marine angelfish pomacanthus*, Nature, 376:6543 (1995), pp. 765–768.

[17] W. Kwan, S. Leung, X. P. Wang, and J. Qian, *A fast huygens sweeping method for capturing paraxial multi-color optical self-focusing in nematic liquid crystals*, J. Comput. Phys., 348 (2017), pp. 108–138.

[18] T. A. M. Langlands and B. I. Henry, *The accuracy and stability of an implicit solution method for the fractional diffusion equation*, J. Comput. Phys., 205 (2005), pp. 719–736.

[19] B. K. Larkin, *Some stable explicit difference approximations to the diffusion equation*, Mathematics of Computation, 18:86 (1964), pp. 196–202.

[20] S. Leung and S. Osher, *Alternating Direction Explicit (ADE) scheme for time-dependent evolution equations*, UCLA CAM Report (11-52), 2005.

[21] S. Leung, J. Qian, and S. Serna, *Fast huygens sweeping methods for schrödinger equations in the semi-classical regime*, Methods and Applications Methods and Applications of Analysis, 2013.

[22] X. Li and C. Xu, *A space-time spectral method for the time fractional difussion equation*, SIAM J. Numer. Anal., 47:3 (2009), pp. 2108–2131.

[23] Y. M. Lin and C. J. Xu, *Finite difference/spectral approximation for the time fractional diffusion equations*, J. Comput. Phys., pp. 1533–1552, 2007.

[24] F. Liu, S. Shen, V. Anh, and I. Turner, *Analysis of a discrete non-Markovian random walk approximation for the time fractional diffusion equation*, ANZIAM J., 46 (2005), pp. C488–C504.

[25] T. Lu, P. Neittaanmaki, and X.-C. Tai, *A parallel splitting up method and its application to*

Navier-Stokes equations, Applied Mathematics Letters, 4 (1991), pp. 25–29.

[26] T. Lu, P. Neittaanmaki, and X.-C. Tai, *A parallel splitting up method for partial differential equations and its application to Navier-Stokes equations*, RAIRO Math. Model. and Numer. Anal., 26 (1992), pp. 673–708.

[27] R. Metzler, J. Klafter, J. Jortner, and M. Volk, *Multiple time scales for dispersive kinetics in early events of peptide folding*, Chemical physics letters, 293:5 (1998), pp. 477–484.

[28] K. S. Miller and B. Ross, *An introduction to the fractional calculus and fractional differential equations*, Wiley-Interscience, New York, 1993.

[29] I. Podlubny, *Fractional differential equations*, Academic Press, Inc., 1998.

[30] S. Sick, S. Reinker, J. Timmer, and T. Schlake, *Wnt and dkk determine hair follicle spacing through a reaction-diffusion mechanism*, Science, 314:5804 (2006), pp. 1447–1450.

[31] P. Smereka, *Semi-implicit level set methods for curvature and surface diffusion motion*, Journal of Scientific Computing, 19:1 (2003), pp. 439–456.

[32] J. C. Strikwerda, *Finite difference schemes and partial differential equations*, Wadsworth-Brooks/Cole, Pacific Grove, California, 1989.

[33] Z. Z. Sun and X. N. Wu, *A fully discrete difference scheme for a diffusion-wave system*, Appl. Num. Math., 56 (2006), pp. 193–209.

[34] A. M. Turing, *The chemical basis of morphogenesis*, Philosophical Transactions of the Royal Society of London B: Biological Sciences, 237:641 (1952), pp. 37–72.

[35] P. N. Vabishchevich, *Additive Operator-Difference Schemes: Splitting Schemes*, De Gruyter, 2013.

[36] W. Wyss, *The fractional diffusion equation*, J. Math. Phys., 27 (1996), pp. 2782–2785.

[37] G. You and S. Leung, *A fast semi-implicit level set method for curvature dependent flows with an application to limit cycles extraction in dynamical systems*, Commun. Comput. Phys., 18:1 (2015), pp. 203–229.