Iterative Operator-Splitting Methods and Continuous and Discrete Case: Theory and Applications

Jürgen Geiser *and Gamze Tanoğlu †
Humboldt Univeristät zu Berlin, D- Berlin, Germany,
Izmir Institute of Technology, Gulbahce Campus, Urla, Izmir,35430, Turkey

June 27, 2009

Abstract

In this paper, we contribute waveform relaxation and iterative splitting methods for systems of parabolic differential equations. We could present an analysis comparing both methods and see advantages in the iterative splitting method.

Here the benefits are combination of large and small time-scales, which one the large time-scale the computational effort is less and on the small time-scale the computational work is tremendous.

We discuss the convergence analysis in the finite and infinite time-interval, see [22].

The applications can be done for parabolic equations with nonlinear parts.

Such problems can be decoupled in two problems, where on the one side the less investigated operator is solved with cheap methods, e.g. implicit Euler methods and the other part with high accurate methods, e.g. Runge-Kutta methods of higher order.

We present the method with comparison to standard Fractional-Stepping methods.

The benefit will be the individual handling of each operators with adapted standard higher order time-integrators. The methods are applied to convection-diffusion-reaction equations as used to model financial options. Finally we discuss the modified methods for multi-dimensional and multi-physical problems.

Keywords. Operator splitting method, Iterative solver method, Stability analysis, Weighting methods.

AMS subject classifications. 80A20, 80M25, 74S10, 76R50, 35J60, 35J65, 65M99, 65Z05, 65N12

1 Introduction

We motivate our studying on combining explicit and implicit time-discretization methods with iterative Operator-Splitting methods as efficient discretization- and solver-methods.

The main advantage is using standard implicit and explicit Runge-Kutta or BDF-method and embed this methods in an iterative solver. We discuss the efficiency to the more complicate family of IMEX-methods, combining explicit and implicit method in one complicate method.

*email: geiser@mathematik.hu-berlin.de
†email: gamzetanoglu@iyte.edu.tr

2 Mathematical Model

Our model equations are coming from a computational simulation of bio-remediation [2] or radioactive contaminants [9], [8].

The mathematical equations are given by

$$\partial_t \, R \, c + \nabla \cdot (\mathbf{v}c - D(c)\nabla c) = f(c) \, , \tag{1}$$

$$f(c) = c^p \, , \text{ chemical-reaction and } p > 0 \tag{2}$$

$$f(c) = \frac{c}{1-c} \, , \text{ bio-remediation} \tag{3}$$

The unknown $c = c(x,t)$ is considered in $\Omega \times (0,T) \subset \mathbb{R}^d \times \mathbb{R}$, the space-dimension is given by d . The Parameter $R \in \mathbb{R}^+$ is constant and is named as retardation factor. The other parameters $f(c)$ are nonlinear functions, for example bio-remediation or chemical reaction. $D(c)$ is the nonlinear diffusion-dispersion tensor and $\mathbf{v}$ is the velocity.

The aim of this paper is to present a new iterative method based on operator-splitting methods for partial differential equations. In a first paper, we focus on ordinary differential equations and discuss the theory and application for a weighted method.

3 Iterative splitting method

In this work, we consider the following form of the Iterative splitting schemes:

1. Iterative splitting with respect to one operator

$$\frac{\partial c_i(t)}{\partial t} = A_1 c_i(t) \; + \; A_2 c_{i-1}(t), \text{ with } \; c_i(t^n) = c^n, i = 1, 2, \ldots, m \tag{4}$$

2. Iterative splitting with respect to alternating operators

$$\begin{aligned} &\frac{\partial c_i(t)}{\partial t} = A_1 c_i(t) \; + \; A_2 c_{i-1}(t), \text{ with } \; c_i(t^n) = c^n \\ &i = 1, 2, \ldots, j \, , \\ &\frac{\partial c_i(t)}{\partial t} = A c_{i-1}(t) \; + \; B c_i(t), \text{ with } \; c_{i+1}(t^n) = c^n \, , \\ &i = j+1, j+2, \ldots, m \, , \end{aligned} \tag{5}$$

3. Unsymmetrical weighted iterative splitting

$$\begin{aligned} &\frac{\partial c_i(t)}{\partial t} = A_1 c_i(t) \; + \; w A_2 c_{i-1}(t), \text{ with } \; c_i(t^n) = c^n \\ &i = 1, 2, \ldots, j \, , \\ &\frac{\partial c_i(t)}{\partial t} = w A_1 c_{i-1}(t) \; + \; A_2 c_i(t), \text{ with } \; c_{i+1}(t^n) = w c^n + (1-w) c_{i+1}(t^{n+1}) \, , \\ &i = j+1, j+2, \ldots, m \, , \end{aligned} \tag{6}$$

4. Symmetrical weighted iterative splitting

$$\begin{aligned}
&\frac{\partial c_i(t)}{\partial t} = 2wA_1c_i(t) \ + \ (1-2w)A_2c_{i-1}(t),\\
&\text{with } \ c_i(t^n) = c^n, \ i = 1, 2, \ldots, j \ ,\\
&\frac{\partial c_i(t)}{\partial t} = (1-2w)A_1c_{i-1}(t) \ + \ wA_2c_i(t),\\
&\text{with } \ c_{i+1}(t^n) = 2wc^n + (1-2w)c_{i+1}(t^{n+1}), \ i = j+1, j+2, \ldots, m,
\end{aligned} \tag{7}$$

where we assume the operator A_1 has a large time scale and A_2 has a small time scale. In addition, $c_0(t^n) = c^n$, $c_{-1} = 0$ and c^n is the known split approximation at the time level $t = t^n$. The split approximation at the time-level $t = t^{n+1}$ is defined as $c^{n+1} = c_{2m+1}(t^{n+1})$. (Clearly, the function $c_{i+1}(t)$ depends on the interval $[t^n, t^{n+1}]$, too, but, for the sake of simplicity, in our notation we omit the dependence on n.)

4 Improved Iterative splitting method

Often standard iterative splitting methods have the problem to be less effective in the convergence or the stability.

Here we propose the two benefits of improving the iterative schemes:

- Initialization (improve the starting conditions)
- Acceleration (Convolution ideas, weighting ideas or Krylov space ideas)

4.1 Improved initialization

The main problem is the initialization.

Often the $c_0(t) = c(t^n)$ or $c_0(t) = 0$ are to fare from the result, see

$$||c - c_0|| \leq err \tag{8}$$

where err is a given starting error.

By the way the standard initialization errors are

$$||c(t) - c_n|| \leq ||(\exp((A+B)t) - I)c_n|| \tag{9}$$

$$||c(t) - 0|| = ||\exp((A+B)t)c_n|| \tag{10}$$

and are of zero order and to large at all.

Here the ideas of prestepping methods,e.g. A-B splitting or Strang splitting as first or second order exponential splitting schemes can reduce the initial error.

See for the A-B splitting we have global a first order scheme

$$c_0(t) = \exp(At)\exp(Bt)c_n, \tag{11}$$

$$||c(t) - c_0(t)|| \leq O(t^2) \tag{12}$$

where for the Strang splitting we have global a second order scheme

$$c_0(t) = \exp(At/)\exp(Bt)\exp(A/2t)c_n, \tag{13}$$

$$||c(t) - c_0(t)|| \leq O(t^3) \tag{14}$$

Remark 4.1 *Here obtain a starting condition of first or second order and can improve with iterative steps to higher order schemes.*

4.2 Improved convergence

Here the main problem is slow convergence, while not using acceleration techniques.

With relaxation schemes, we can obtain faster schemes.

4.2.1 Weighted iterative operator splitting methods

Here the idea are based on using the inverse function to accelerate the solver process:

$$\begin{aligned}
&\frac{\partial c_i(t)}{\partial t} = A_1 c_i(t) \ + \ A_2 c_{i-1}(t), \\
&\text{with } \ c_i(t^n) = \omega c^n + (1-\omega)\exp(-(A_1+A_2)\tau) c_{i-1}(t^{n+1}), \ i = 1, 2, \ldots, j \ , \\
&\frac{\partial c_i(t)}{\partial t} = A_1 c_{i-1}(t) \ + \ A_2 c_i(t), \\
&\text{with } \ c_{i+1}(t^n) = \omega c^n + (1-\omega)\exp(-(A_1+A_2)\tau) c_i(t^{n+1}), \ i = j+1, j+2, \ldots, m, \quad (15)
\end{aligned}$$

where we assume the operator A_1 has a large time scale and A_2 has a small time scale. In addition, $c_0(t)$ is computed with an exponential splitting scheme.

Here the underlying idea is based on the assumption, that we can simple compute the inverse of the solution, means:

$$c_n = \exp(-(A+B)\tau) c(t), \tag{16}$$

where $\tau = t - t_n$ and $c(t)$ is the exact solution.

Iteratively the residual is given as:

$$c_n - \exp(-(A+B)\tau) c_i(t) = r_i(t), \tag{17}$$

where c_i is the i-th iterative solution. So we have to correct the initial values if the residual are large.

So we have to balance with a weighting function ω.

In the following theorem we discuss the improved method for the initial value problem:

$$\partial_t c(t) = Ac(t) + Bc(t) \quad t \in (0,T), \quad c(0) = c_0, \tag{18}$$

where the initial function c_0 is given, and A and B are assumed to be bounded linear operators. (In many application the operator A, B came from finite difference or finite volume discretization, e.g. they correspond to the discretised in space convection and diffusion operators.)

Classical A-B splitting method

Theorem 4.2 *We have to solve the initial value problem with a classical $A-B$ splitting method. Th splitting error of the $A-B$ method is given as (see [16]):*

$$\begin{aligned}
\rho_n &= \frac{1}{\tau_n}(\exp(\tau_n(A+B)) - \exp(\tau_n B)\exp(\tau_n A))\ c(t^n) \\
&= \frac{1}{2}\tau_n [A,B]\ c(t^n) + \mathcal{O}(\tau_n^2), \qquad (19)
\end{aligned}$$

where $[A,B] := AB - BA$ is the commutator of A and B. Consequently, the splitting error is $O(\tau_n)$ when the operators A and B do not commute.

We improve the method by assuming:
$\exp(-(A+B)t)$ *can be computed simple.*
We apply additional the step:

$$\tilde{c}_{n+1} = c_{n+1,AB} - (1 - \exp(-(A+B)t)c_{n+1,AB})c_n \tag{20}$$
$$\tag{21}$$

and we obtain a higher order result:

$$\tilde{\rho}_n \quad = \quad \mathcal{O}(\tau_n^2), \tag{22}$$

In the next we discuss the proof.

Proof 4.1 *We have the following contribution:*

$$c_{n+1,AB} \quad = \quad \exp(Bt)\exp(At)c^n, \tag{23}$$

The inverse step is given as:

$$\tilde{c}_n = \exp(-(A+B)t)\exp(Bt)\exp(At)c^n. \tag{24}$$

We obtain the corrected solution:

$$\tilde{c}_{n+1} = c_{n+1,AB} - (1 - \exp(-(A+B)t)\exp(Bt)\exp(At))c_n \tag{25}$$

and

$$||c(t_{n+1}) - \tilde{c}_{n+1}|| \tag{26}$$
$$= ||\exp((A+B)t)c_n - (c_{n+1,AB} - (1 - \exp(-(A+B)t)\exp(Bt)\exp(At)))c_n|| \tag{27}$$
$$||[A,B]t^2/2 + O(t^3) - [A,B]t^2/2 - O(t^3)||||c_n|| \tag{28}$$
$$O(t^3)||c_n||, \tag{29}$$

and globally we obtain $O(t^2)$.

The same idea can be obtained with the iterative splitting method.
Iterative splitting methods
We have to solve the initial value problem with an iterative splitting method.

Theorem 4.3 *The splitting error of the iterative method is given as (see [7]):*

$$\rho_{n,iter,i} \quad = \quad \frac{1}{i!}(\tau_n^i)(A+B)^i\, c(t^n) + \mathcal{O}(\tau_n^{i+1}) = \mathcal{O}(\tau_n^i), \tag{30}$$

where we have operators A *and* B *do not commute.*
We improve the method by assuming:
$\exp(-(A+B)t)$ *can be computed simple.*
We apply additional the step:

$$\tilde{c}_{n+1,iter,i} = c_{n+1,iter,i} - (1 - \exp(-(A+B)t)c_{n+1,iter,i})c_n \tag{31}$$
$$\tag{32}$$

and we obtain a higher order result:

$$\rho, \tilde{iter}, i_n \quad = \quad \mathcal{O}(\tau_n^{i+1}), \tag{33}$$

Proof 4.2 *The same proof idea as in the $A-B$ splitting.*

Remark 4.4 *Numerically we can apply the method by weighting factors, while we often only obtain the minimal and maximal eigenvalues of the operators. Here we apply the weighted methods as acceleration methods, based on the inverse idea to skip the error term and obtain one order more.*

4.2.2 Accelerated iterative operator splitting methods

The convergence rate of iterative operator splitting methods are often very slow for many problems of interest.

CSOR Method

As with relaxation-based approaches for linear algebra (e.g. Gauss-Jacobi), application of appropriate acceleration is necessary to make the iterative approach practical, see [**?**].

The ideas are the following here we combine Gauss Seidel WR with CSOR acceleration:

We have the following problem:

$$\frac{du}{dt} = Au,\ t \in [t^n, t^{n+1}], \tag{34}$$

where $A = D + L + U$,

The iterative scheme is given as:

$$\frac{du_i}{dt} = Du_i + L\hat{u}_i + Uu_{i-1}, \tag{35}$$

where $\hat{u}_i(t) = u_{i-1}(t) + \int_0^t \omega(t)(\hat{u}_i(t-\tau) - u_i(t-\tau))d\tau$.

We iterate $i = 1, 2, \ldots$, and initialization $u_0 = u(t_n)$.

Krylov subspaces combined with iterative operator splitting scheme

Here we assume the following problem of our splitting scheme: The inverse solution $\exp(At)^{-1}$ is simple to compute.

We can rewrite the equation:

$$\frac{du_i}{dt} = A/nu_i + \ldots + A/nu_{i-1}, \tag{36}$$

where $i = 1, 2, \ldots$ and obtain the solution u_i

$$\exp(-At)u_i dt - u(t^n) = r_i, \tag{37}$$

this is the residual, where the exact solution is $\exp(-At)u(t) = c_n$.

We apply the CCG algorithm, e.g. 2 times with

Algorithm 4.5 *1.) Pick the smooth $u_i = u_{i,0}$ and compute $r_j = c_n - \exp(-At)u_{i,0}$*

2.) For $j = 0, 1, 2, 3, ldots, J$ do

$$\alpha_j = (r_j \cdot r_j)/(\exp(-At)p_j \cdot p_j) \tag{38}$$
$$u_{i,j+1} = u_{i,j} + \alpha_j \cdot p_j \tag{39}$$
$$r_{j+1} = r_j - \alpha_j \cdot \exp(-At)p_j \tag{40}$$
$$\beta_j = (r_{j+1} \cdot r_{j+1})/(r_j \cdot r_j) \tag{41}$$
$$p_{j+1} = r_{j+1} + \beta \cdot p_j \tag{42}$$
$$\tag{43}$$

3.) Do $\tilde{u}_i = u_{i,J}$. and start the iterative scheme (36) to solve u_{i+1}.

5 Stability Analysis

In this section, we derive the stability function after applying the BDFk and SBDFk, where k denotes the order of the method, to each sub-equations of the iterative operator splitting given in Equations (5), (6) and (15).

5.1 Construction of the BDFk methods:

In literature, BDF is given by the equation,

$$\sum_{r=1}^{k+1} \alpha_r u^{n-i+2} = \tau\beta f(u^{n+1}). \tag{44}$$

where the coefficients α_r and β are obtained based on the Taylor expansion and difference operator. We construct the BDFk method as follows,

$$\sum_{r=1}^{k+1} \alpha_r u^{n+1} = \tau\beta A_2 u^{n+1} + \tau\beta A_2 u^{n+1}, \tag{45}$$

and the following conditions must be satisfy:

1. $k < 7$, otherwise the method is not zero stable,
2. $\sum_{r=1}^{k+1} \alpha_r = 0$, otherwise the method is not consistent,

5.2 Construction of the SBDFk methods:

We would like to design k-th order SBDF method as follows:

$$\sum_{r=1}^{k+1} \alpha_r u^{n-i+2} = \tau\beta_1 A_2 u^{n+1} + \tau \sum_{r=2}^{k+1} \beta_r A_1 u^{n-i+2}, \tag{46}$$

the following conditions must be satisfy:

1. $k < 7$, otherwise the method is not zero stable,
2. $\sum_{r=1}^{k+1} \alpha_r = 0$, otherwise the method is not consistent,
3. $\beta_1 = \sum_{r=2}^{k+1} \beta_r$, otherwise the method is not consistent.

In the Equation (46), the parameters $\alpha_i, i = 1...., k+1$ are the same as BDFk method. One can rescale the the Equation (46) by dividing β_1 both sides of the equation , thus the unknown parameters $\beta_i, i = 2...., k+1$ can be found by solving linear system of equation obtained by Taylor expansions of dependent variables around u^n as follows:

For k is even:

$$\sum_{r=1}^{k+1} \alpha_r u^{n-i+2} = \tau A_2 u^{n+1} + \tau \sum_{r=1}^{k} (-1)^{k+r-1} \beta_r \begin{pmatrix} k \\ r \end{pmatrix} A_1 \tag{47}$$

For k is odd:

$$\sum_{r=1}^{k+1} \alpha_r u^{n-i+2} = \tau A_2 u^{n+1} + \tau \sum_{r=1}^{k} (-1)^{k+r-2} \beta_r \begin{pmatrix} k \\ r \end{pmatrix} A_1 \tag{48}$$

These linear system can be written in the closed form by choosing the parameter $\beta_1 = 1$, as follows:

$$\sum_{s=1}^{k} \begin{pmatrix} k \\ s \end{pmatrix} t_s = 1 \text{ for } i = 1 \tag{49}$$

$$\sum_{s=2}^{k} \begin{pmatrix} k \\ s \end{pmatrix} (-1)^{i+1} t_s (s-1)^{i-1} = 1 \text{ for } i = 2, \ldots, k \tag{50}$$

where $\beta_r = t_{r-1} \begin{pmatrix} k \\ r-1 \end{pmatrix}$.

For example, by using the formulas given in (49) and (50), the system of equations need to solved are

For k=2,

$$\begin{pmatrix} k & \begin{pmatrix} k \\ k-2 \end{pmatrix} \\ 0 & -\begin{pmatrix} k \\ k-2 \end{pmatrix} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{51}$$

For k=3,

$$\begin{pmatrix} k & \begin{pmatrix} k \\ k-2 \end{pmatrix} & 1 \\ 0 & -\begin{pmatrix} k \\ k-2 \end{pmatrix} & -(k-1) \\ 0 & \begin{pmatrix} k \\ k-2 \end{pmatrix} & (k-1)^2 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{52}$$

For $k = 4$

$$\begin{pmatrix} k & \begin{pmatrix} k \\ k-2 \end{pmatrix} & 1 & k \\ 0 & -\begin{pmatrix} k \\ k-2 \end{pmatrix} & -(k-1) & -k(k-2) \\ 0 & \begin{pmatrix} k \\ k-2 \end{pmatrix} & (k-1)^2 & k(k-2)^2 \\ 0 & -\begin{pmatrix} k \\ k-2 \end{pmatrix} & -(k-1)^3 & -k(k-2)^3 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \tag{53}$$

For $k = 5$

$$\begin{pmatrix} k & \binom{k}{k-2} & 1 & k & k(k-3) \\ 0 & -\binom{k}{k-2} & -(k-1) & -k(k-2) & -k(k-3)^2 \\ 0 & \binom{k}{k-2} & (k-1)^2 & k(k-2)^2 & k(k-3)^3 \\ 0 & -\binom{k}{k-2} & -(k-1)^3 & -k(k-2)^3 & -k(k-3)^4 \\ 0 & \binom{k}{k-2} & (k-1)^4 & k(k-2)^4 & k(k-3)^5 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad (54)$$

5.3 Stability Function I

In this section, we derive the stability of iterative operator splitting method on the time interval $[t^n, t^{n+1}]$, we solve the each sub-problems given in Equations (5), (6), and (15) consecutively by applying BDFk and SBDFk methods. We can generalize the stability with respect to the following Theorem:

Theorem 5.1 *We apply BDF- and SBDF- methods to the iterative operator splitting method given in Equation (5), then for the linear system with $Z_1 = \tau A_1$ and $Z_2 = \tau A_2$ we obtain stabile functions $R_i(Z_1, Z_2)$ with:*

$$||R_i(Z_1, Z_2)|| \leq 1, \quad (55)$$
$$\text{for all } Z_1, Z_2 \in \mathbf{X} \times \mathbf{X}, \quad (56)$$

where $\mathbf{X}$ is a Banach-space and $||\cdot||$ a matrix norm.

Proof 5.1 *The proof the techniques are done based on the BDF and the new design SBDFk, stiff backward differential formulas,*

1a) Finite time-intervals with Standard BDFk method:

Discretization of the Equations (5) by BDFk methods are given by

$$(\alpha_1 I - \beta\tau A_1)c_i^{n+1} = -\sum_{i=2}^{k+1} \alpha_k c_i^{n-i+2} + \beta\tau A_2 c_{i-1}^{n+1}, i = 1, 2, ..j \quad (57)$$

$$(\alpha_1 I - \beta\tau A_2)c_{i+1}^{n+1} = -\sum_{i=2}^{k+1} \alpha_k c_i^{n-i+2} + \beta\tau A_1 c_i^{n+1}, i = 1, 2, ..j, i = j+1, ...m \quad (58)$$

For the linear system we denote $Z_1 = \tau A_1$ and $Z_2 = \tau A_2$ and we set $c_i^n = c_{i+1}^n = c_i^{n-1} = c_{i+1}^{n-1} = ... = c_i^{n-k+1} = c_{i+1}^{n-k+1} = c^n$ and initialize with $c_0^{n+1} = c^n$ and use the conditions 2, we have We get the following stability equation, cf. [Hundsdorfer 2005]: We compute the first iteration with $i = 1$ and get the equation

$$
\begin{aligned}
(\alpha_1 I - \beta Z_1)c_1^{n+1} &= -\sum_{i=2}^{k+1} \alpha_i c^n + \beta Z_2 c^n, \;\; i = 1, 2, ..j \\
&= \alpha_1 c^n + \beta Z_2 c^n, \;\; i = 1, 2, ..j \\
&= (\alpha_1 I + \beta Z_2)c^n, \;\; i = 1, 2, ..j \\
c_1^{n+1} &= (I - \frac{\beta}{\alpha_1} Z_1)^{-1}(\; I + \frac{\beta}{\alpha_1} Z_2))c^n \qquad (59) \\
c_1^{n+1} &= R_1(Z_1, Z_2)c^n \qquad (60)
\end{aligned}
$$

where we define

$$
\begin{aligned}
R_1(Z_1, Z_2) &= R_{(-Z_1)}^{-1} R_{(+Z_2)} \qquad (61) \\
R_\theta &= (I + a_k \theta) \qquad (62) \\
a_1 &= \frac{\beta}{\alpha_1} \qquad (63)
\end{aligned}
$$

For next iteration i=2, we get the stability function same as before as before

$$
\begin{aligned}
(\alpha_1 I - \beta Z_2)c_2^{n+1} &= -\sum_{i=2}^{k+1} \alpha_i c^n + \beta Z_1 {c_1}^{n+1}, \;\; i = 1, 2, ..j \\
&= \alpha_1 c^n + \beta Z_1 R_1(Z_1, Z_2)c^n, \;\; i = 1, 2, ..j \\
&= (\alpha_1 I + \beta Z_1 R_1(Z_1, Z_2))c^n, \;\; i = 1, 2, ..j \\
c_2^{n+1} &= (I - a_k Z_2)^{-1}(\; I + a_k R_1(Z_1, Z_2)Z_1)c^n, \;\; i = 1, 2, ..j \qquad (64) \\
c_2^{n+1} &= R_2(Z_1, Z_2)c^n \qquad (65)
\end{aligned}
$$

where we define,

$$
R_2(Z_1, Z_2) = R_{(-Z_2)}^{-1} R_{(+R_1 Z_1)} \qquad (66)
$$

We may generalize the result recursively as follows:

For p is odd:

$$
\begin{aligned}
R_0(Z_1, Z_2) &= I \qquad (67) \\
R_p(Z_1, Z_2) &= R_{(-Z_1)}^{-1}(\; I + a_k R_{p-1} Z_2), \quad p = 1, 3.. \qquad (68)
\end{aligned}
$$

For p is even:

$$
R_{2p+2}(Z_1, Z_2) = R_{(-Z_2)}^{-1}(\; I + a_k R_{2p+1} Z_1), \quad p = 0, 2.. \qquad (69)
$$

Thus, the solutions of the sub equations can be written in terms of the stability function as follows:

$$
c_i = R_i(Z_1, Z_2)c^n, \quad i = 1, 2, 3...m \qquad (70)
$$

These results can also written in terms of the initial conditions:

$$
c^n = R_i(Z_1, Z_2)^n c^0, \quad i = 1, 2, 3...m \qquad (71)
$$

1b.)Finite time-intervals with SBDFk method :

Discretization of the Equations (5) by SBDFk methods are given by

$$\alpha_1 c_i^{n+1} = \sum_{i=2}^{k+1}(\beta_i \tau A_1 - \alpha_i I)c_i^{n-i+2} + \beta_1 \tau A_2 c_{i-1}^{n+1}, i = 1, 2, ..j \tag{72}$$

$$(\alpha_1 I - \beta_1 \tau A_2)c_{i+1}^{n+1} = \sum_{i=2}^{k+1}(\beta_i \tau A_1 - \alpha_i I){c_i}^{n-i+2}, i = j+1, ...m \tag{73}$$

For the linear system we denote $Z_1 = \tau A_1$ and $Z_2 = \tau A_2$ and we set $c_i^n = c_{i+1}^n = c_i^{n-1} = c_{i+1}^{n-1} = ... = c_i^{n-k+1} = c_{i+1}^{n-k+1} = c^n$ and initialize with $c_0^{n+1} = c^n$.

We get the following stability equation, cf. [Hundsdorfer 2005]: We compute the first iteration with $i = 1$ and get the equation

$$\begin{aligned}
\alpha_1 c_1^{n+1} &= \sum_{i=2}^{k+1}(\beta_i Z_1 - \alpha_i I)c^n + \beta_1 Z_2 c^n, \ \ i = 1, 2, ..j \\
&= [(\sum_{i=2}^{k+1}\beta_i)Z_1 - \sum_{i=2}^{k+1}\alpha_i)]c^n + \beta_1 Z_2 c^n, \ \ i = 1, 2, ..j \\
&= (\beta_1 Z_1 + \alpha_1)c^n + \beta_1 Z_2 c^n, \ \ i = 1, 2, ..j \\
c_1^{n+1} &= (I + \frac{\beta_1}{\alpha_1}(Z_1 + Z_2)c^n, \ \ i = 1, 2, ..j
\end{aligned} \tag{74}$$

by using the conditions 2 and 3.

For next iteration i=2, we get

$$\begin{aligned}
(\alpha_1 I - \beta_1 Z_2)c_2^{n+1} &= \sum_{i=2}^{k+1}(\beta_i Z_1 - \alpha_i I)c^n, \ \ i = j+1, ...m \\
&= (\beta_1 Z_1 + \alpha_1 I)c^n, \ \ i = j+1, ...m \\
c_2^{n+1} &= (I - \frac{\beta_1}{\alpha_1}Z_2)(I + \frac{\beta_1}{\alpha_1}Z_1)c^n, \ \ i = j+1, ...m
\end{aligned} \tag{75}$$

$$\tag{76}$$

Hence, we will get the following stability function for iterative splitting after applying the k-th SBDF method,

$$c_1^{n+1} = R_1(Z_1, Z_2)c^n \tag{77}$$

where we define,

$$R_1(Z_1, Z_2) = I + a_k(Z_1 + Z_2)c^n$$

and $a_k = \frac{\beta_1}{\alpha_1}$ is fixed and satisfied $0 < a_k < 1$.

For the second iteration we have

$$c_2^{n+1} = (I - a_k Z_2)^{-1}(I + a_k Z_1)c^n \tag{78}$$

We have stability equation

$$c_2^{n+1} = R_{-Z_2}^{-1} R_{+Z_1} c^n \quad (79)$$
$$c_2^{n+1} = R_2(Z_1, Z_2) c^n \quad (80)$$

with the prestep we obtain the stable function: $||\tilde{R}_2(Z_1, Z_2)|| = ||R_2(Z_1, Z_2)(I-(1-a_k)Z_1)^{-1}|| \leq 1$ where k is the k-th SBDF method, $a_k = \frac{\beta_1}{\alpha_1}$ is fixed and $0 < a_k \leq 1$,.

We may generalize the result recursively as follows:

For p is odd:

$$R_0(Z_1, Z_2) = I \quad (81)$$
$$R_{2p+1}(Z_1, Z_2) = R_{(Z_1+R_{2p}Z_2)} \quad p = 0, 1, 2.. \quad (82)$$

For p is even:

$$R_{2p+2}(Z_1, Z_2) = R_{(-Z_2)}^{-1} R_{(Z_1)}, \quad p = 0, 1, 2.. \quad (83)$$

Thus, the solutions of the sub equations can be written in terms of the stability function as follows:

$$c_i = R_i(Z_1, Z_2) c^n, \quad i = 1, 2, 3...m \quad (84)$$

These results can also written in terms of the initial conditions:

$$c^n = R_i(Z_1, Z_2)^n c^0, \quad i = 1, 2, 3...m \quad (85)$$

Theorem 5.2 *We apply BDFk- and SBDFk- methods to the iterative operator splitting method given in Equation (6), then for the linear system with $Z_1 = \tau A_1$ and $Z_2 = \tau A_2$ we obtain stabile functions $R_i(Z_1, Z_2, w)$ with:*

$$||R_i(Z_1, Z_2, w)|| \leq 1, \quad (86)$$
$$\text{for all } Z_1, Z_2 \in \mathbf{X} \times \mathbf{X}, \quad (87)$$

where $\mathbf{X}$ is a Banach-space and $||\cdot||$ a matrix norm.

Proof 5.2 *The proof the techniques are the same as before for SBDFk method. After the small modification of the Theorem 5.1, we obtain following stability function:*

For odd iteration:

$$c_{2i-1}^{n+1} = R_1(Z_1, Z_2, w) c^n \quad (88)$$

where we define,

$$R_1(Z_1, Z_2, w) = I + a_k(Z_1 + wZ_2)$$

and $a_k = \frac{\beta_1}{\alpha_1}$ is fixed and satisfied $0 < a_k < 1$.

For even iteration:

$$c_{2i}^{n+1} = ((I - a_k Z_2)^{-1}(I + a_k w Z_1)(w + (1 - w)R_1(Z_1, Z_2, w))c^n \quad (89)$$

We have stability equation

$$\begin{aligned} c_{2i}^{n+1} &= R_-(Z_2)^{-1}R_+(Z_1,w)(w+(1-w)R_1(Z_1,Z_2,w))c^n \quad (90)\\ c_{2i}^{n+1} &= R_k(Z_1,Z_2,w)(w+(1-w)R_1(Z_1,Z_2,w))c^n \quad (91)\end{aligned}$$

with the prestep we obtain the stable function:

$$||\tilde{R}_k(Z_1,Z_2,w)|| = ||R_k(Z_1,Z_2,w)(I-(1-wa_k)Z_1)^{-1}(w+(1-w)R_1(Z_1,Z_2,w))|| \leq 1$$

where k is the k-th SBDF method, $a_k = \frac{\beta_1}{\alpha_1}$ is fixed and $0 < a_k \leq 1$,.

Theorem 5.3 *We apply BDFk- and SBDFk- methods to the iterative operator splitting method given in Equation (15), then for the linear system with $Z_1 = \tau A_1$ and $Z_2 = \tau A_2$ we obtain stabile functions $R_i(Z_1,Z_2,w)$ with:*

$$||R_i(Z_1,Z_2,w)|| \leq 1, \quad (92)$$
$$\text{for all } Z_1,Z_2 \in \mathbf{X}\times\mathbf{X}, \quad (93)$$

where $\mathbf{X}$ is a Banach-space and $||\cdot||$ a matrix norm.

Proof 5.3 *The proof the techniques are the same as before for SBDFk method. After the small modification of the Thm5.1, we obtain following stability function:*

For odd iteration:

$$c_{2i-1}^{n+1} = R_1(Z_1,Z_2,w)c^n \quad (94)$$

where we define,

$$R_1(Z_1,Z_2,w) = I + a_k(2wZ_1 + (1-2w)Z_2)$$

and $a_k = \frac{\beta_1}{\alpha_1}$ is fixed and satisfied $0 < a_k < 1$.

For even iteration:

$$c_{2i}^{n+1} = ((I-a_k(1-2w)Z_2)^{-1}(I+2wa_kZ_1)(2w+(1-2w)R_1(Z_1,Z_2,w))c^n \quad (95)$$

We have stability equation

$$\begin{aligned} c_{2i}^{n+1} &= R_-(Z_2,w)^{-1}R_+(Z_1,w)(2w+(1-2w)R_1(Z_1,Z_2,w))c^n \quad (96)\\ c_{2i}^{n+1} &= R_k(Z_1,Z_2,w)(w+(1-w)R_1(Z_1,Z_2,w))c^n \quad (97)\end{aligned}$$

with the prestep we obtain the stable function:

$$||\tilde{R}_k(Z_1,Z_2,w)|| = ||R_k(Z_1,Z_2,w)(I-(1-wa_k)Z_1)^{-1}(2w+(1-2w)R_1(Z_1,Z_2,w))|| \leq 1$$

where k is the k-th SBDF method, $a_k = \frac{\beta_1}{\alpha_1}$ is fixed and $0 < a_k \leq 1$,.

6 Higher Order Iterative Splitting Based on the Extrapolation for Two Dimensional Problem

In general formulation, we consider an ODEs system

$$\frac{dU}{dt} = (A_1 + A_2)U \ , \tag{98}$$

$$\text{with } U(0) = U^0, \tag{99}$$

where $U(t)$ is an N-dimensional vector,

$$A_1 = \frac{\partial^2}{\partial x^2} \quad \text{and} \quad A_2 = \frac{\partial^2}{\partial y^2}$$

Iterative Splitting method for two dimensional problem is given as follows:

$$\frac{dU_i}{dt} = A_1 U_i + A_2 U_{i-1} \ , \tag{100}$$

$$\text{with } U_i(0) = U^0, \tag{101}$$

$$\frac{dU_{i+1}}{dt} = A_1 U_i + A_2 U_{i+1} \ , \tag{102}$$

$$\text{with } U_{i+1}(0) = U^0, \tag{103}$$

where $i = 1, 3, 5, \ldots$, and $U_0(t) = 0$ (initialization of the scheme). The implicit trapezoidal method applied to Equations (153) and (155) are written

$$[I - \frac{\triangle t}{2} A_1] U_k^{n+1} = [I + \frac{\triangle t}{2} A_1] U_k^n + \frac{\triangle t}{2} A_2 (U_{k-1}^n + U_{k-1}^{n+1}), \tag{104}$$

$$[I - \frac{\triangle t}{2} A_2] U_{2k}^{n+1} = [I + \frac{\triangle t}{2} A_2] U_{2k}^n + \frac{\triangle t}{2} A_1 (U_{2k-1}^n + U_{2k-1}^{n+1}) \tag{105}$$

where $k = 1, 3, 5, \ldots$. Corresponding to space discretization of 2D heat equation using central differences formula,

$$(A_1 U_k)_{i,j} = \frac{1}{\triangle x^2} [U_{k_{(i-1,j)}} - 2U_{k_{(i,j)}} + U_{k_{(i+1,j)}}], \tag{106}$$

$$(A_2 U_k)_{i,j} = \frac{1}{\triangle y^2} [U_{k_{(i,j-1)}} - 2U_{k_{(i,j)}} + U_{k_{(i,j+1)}}], \tag{107}$$

After inserting to the Equations (157) and (158) into the Equation (104), the following system of equations can be obtained for fixed i,

$$(1 + 2s) U_k^{n+1}{}_{(i,j)} - s U_k^{n+1}{}_{(i-1,j)} - s U_k^{n+1}{}_{(i+1,j)} - c1(i,j) = b1(j), \tag{108}$$

where $s = \frac{\triangle t}{2 \triangle x^2}$ $j = 1, \ldots, N_y$, $b1$ is an N_y-dimensional vector with components

$$b1(j) = (1 - 2s) U_k^n{}_{(i,j)} + s U_k^n{}_{(i-1,j)} + s U_k^n{}_{(i+1,j)}, \tag{109}$$

and $c1(i,:)$ is an Ny dimensional vector with components

$$c1(i,j) = (-2e)(U^n_{k-1} + U^{n+1}_{k-1})_{(i,j)} + e(U^n_{k-1} + U^{n+1}_{k-1})_{(i,j-1)} + e(U^n_{k-1} + U^{n+1}_{k-1})_{(i,j+1)} \quad (110)$$

$j = 1, \ldots, Ny$,
where $e = \frac{\triangle t}{2\triangle y^2}$.
Similarly the Equation (105) can be written as a system of equation as follows

$$(1+2e)U^{n+1}_{k\,(i,j)} - eU^{n+1}_{k\,(i,j-1)} - eU^{n+1}_{k\,(i,j+1)} - c2(i,j) = b2(j), \quad (111)$$

where $j = 1, \ldots, Ny$, $b2$ is an Ny dimensional vector with components

$$b2(j) = (1-2e)U^n_{k\,(i,j)} + eU^n_{k\,(i,j-1)} + eU^n_{k\,(i,j+1)}, \quad (112)$$

and $c2(i,:)$ is an Ny dimensional vector with components

$$c2(i,j) = (-2s)(U^n_{k-1} + U^{n+1}_{k-1})_{(i,j)} + s(U^n_{k-1} + U^{n+1}_{k-1})_{(i-1,j)} + s(U^n_{k-1} + U^{n+1}_{k-1})_{(i+1,j)} \quad (113)$$

Proposition 1. *Iterative splitting based on the extrapolation in* $[0, \triangle t]$ *for two dimensional problem is given in the next steps.*

- Step 1 (first $\frac{\triangle t}{2}$ step) , $\alpha = 1/2$:

$$[I - \alpha\frac{\triangle t}{2}A_1]U^{n+1/2}_k = [I + \alpha\frac{\triangle t}{2}A_1]U^n_k + \alpha\frac{\triangle t}{2}A_2(U^n_{k-1} + U^{n+1/2}_{k-1}), \quad (114)$$

$$[I - \alpha\frac{\triangle t}{2}A_2]U^{n+1/2}_{2k} = [I + \alpha\frac{\triangle t}{2}A_2]U^n_{2k} + \frac{\alpha\triangle t}{2}A_1(U^n_{2k-1} + U^{n+1/2}_{2k-1}) \quad (115)$$

  with $U^{n+1/2}_k = U^0_k$ and $U_0 = 0$, for $k = 1, 2, 3, \ldots$.

- Step 2 ($\triangle t$ step) , $\alpha = 1$:

$$[I - \alpha\frac{\triangle t}{2}A_1]\tilde{U}^{n+1}_k = [I + \alpha\frac{\triangle t}{2}A_1]U^{n+1/2}_k + \alpha\frac{\triangle t}{2}A_2(U^{n+1/2}_{k-1} + U^{n+1}_{k-1}), \quad (116)$$

$$[I - \alpha\frac{\triangle t}{2}A_2]\tilde{U}^{n+1}_{2k} = [I + \alpha\frac{\triangle t}{2}A_2]U^{n+1/2}_{2k} + \frac{\alpha\triangle t}{2}A_1(U^{n+1/2}_{2k-1} + U^{n+1}_{2k-1}) \quad (117)$$

- Step 3 ($\triangle t$ step) , $\alpha = 1$:

$$[I - \alpha\frac{\triangle t}{2}A_1]\tilde{\tilde{U}}^{n+1}_k = [I + \alpha\frac{\triangle t}{2}A_1]U^n_k + \alpha\frac{\triangle t}{2}A_2(U^n_{k-1} + U^{n+1}_{k-1}), \quad (118)$$

$$[I - \alpha\frac{\triangle t}{2}A_2]\tilde{\tilde{U}}^{n+1}_{2k} = [I + \alpha\frac{\triangle t}{2}A_2]U^n_{2k} + \alpha\frac{\triangle t}{2}A_1(U^n_{2k-1} + U^{n+1}_{2k-1}) \quad (119)$$

  with $U^{n+1}_k = U^0_k$ and $U_0 = 0$, for $k = 1, 2, 3, \ldots$.

Resulting steps:

$$U^{n+1}_k = 4/3\tilde{\tilde{U}}^{n+1}_k - 1/3\tilde{U}^{n+1}_k \quad (120)$$

6.1 SBDF-Method as improved time-discretization methods

1.) We first apply the SBDF3 method:
Consider the SBDF3-method

$$(1/\triangle t)(\frac{11}{6}u^{n+1} - 3u^n + \frac{3}{2}u^{n-1} - \frac{1}{3}u^{n-2}) \\ = 3A_1(u^n) - 3A_1(u^{n-1}) + A_1(u^{n-2}) + A_2(u^{n+1}) \tag{121}$$

after applying this method to the Equations (153-155) instead of implicit trapezoidal rule, we obtain following sub equations:

$$(\frac{11}{6})(u_k^{n+1}) = (3I + 3\triangle tA_1)(u_k^n) - (3I/2 + 3\triangle tA_1)(u_k^{n-1}) + (I/3 + \triangle tA_1)(u_k^{n-2}) + \triangle tA_2(u_{k-1}^{n+1}) \tag{122}$$

$$(\frac{11}{6}I - \triangle tA_2)(u_{k+1}^{n+1}) = (3I + 3\triangle tA_1)(u_k^n) - (3I/2 + 3\triangle tA_1)(u_k^{n-1}) + (I/3 + \triangle tA_1)(u_k^{n-2}) \tag{123}$$

where $k = 1, 3, 5, \ldots$. After inserting to the Equations (157) and (158) into the Equation (122), the following system of equations can be obtained for fixed index i,

$$(\frac{11}{6}){u_k^{n+1}}_{(i,j)} \quad = \quad b1(j) - b2(j) + b3(j) + b4(j) \tag{124}$$

where $j = 1, \ldots, N_y$, $b1, b2, b3, b4$ are N_y-dimensional vectors with components

$$b1(j) \quad = \quad (3 - 6s)u^n_{k\,(i,j)} + 3s(u^n_{k\,(i-1,j)} + u^n_{k\,(i+1,j)}) \tag{125}$$

$$b2(j) \quad = \quad (3/2 - 6s){u_k^{n-1}}_{(i,j)} + 3s({u_k^{n-1}}_{(i-1,j)} + {u_k^{n-1}}_{(i+1,j)}) \tag{126}$$

$$b3(j) \quad = \quad (1/3 - 2s){u_k^{n-2}}_{(i,j)} + s({u_k^{n-2}}_{(i-1,j)} + {u_k^{n-2}}_{(i+1,j)}) \tag{127}$$

$$b4(j) \quad = \quad -2s(u^{n+1}_{(k-1)(i,j)}) + s(u^{n+1}_{(k-1)(i,j-1)} + u^{n+1}_{(k-1)(i,j+1)}) \tag{128}$$

where $s = \frac{\triangle t}{\triangle x^2}$.

Similarly the Equation (123) can be written as a system of equation as follows

$$(\frac{11}{6} + 2e)u^{n+1}_{(k+1)}(i,j) - e(u^{n+1}_{(k+1)}(i,j-1) + u^{n+1}_{(k+1)}(i,j+1)) \quad = \quad c1(i,j) \tag{129}$$

where $i = 1, \ldots, Nx$,$j = 1, \ldots, Ny$ and $e = \frac{\triangle t}{\triangle y^2}$.
For fixed i,the vector

$$u^{n+1}_{(k+1)}(i,:)$$

may be found by solving the tridiagonal system

$$Au^{n+1}_{(k+1)}(i,:) = \sum_{m=1}^{3}(-1)^{m+1}bm(j)$$

where the matrix A is tridiagonal,

$$\begin{pmatrix} \frac{11}{6}+2e & -e & 0 & 0 & 0 & 0 & 0 \\ -e & \frac{11}{6}+2e & -e & 0 & 0 & 0 & 0 \\ 0 & -e & \frac{11}{6}+2e & -e & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & 0 & 0 \\ 0 & 0 & 0 & -e & \frac{11}{6}+2e & -e & 0 \\ 0 & 0 & 0 & 0 & -e & \frac{11}{6}+2e & -e \\ 0 & 0 & 0 & 0 & 0 & -e & \frac{11}{6}+2e \end{pmatrix}$$

and bm are Ny dimensional vectors ,$m = 1, 2, 3$, $j = 1, \ldots, Ny$
and the algorithm is given below

for $i = 1 : N_x$
for $j = 1 : N_y$

$$\begin{aligned} b1(j) &= (3-6s)u^n_{k\,(i,j)} + 3s(u^n_{k\,(i-1,j)} + u^n_{k\,(i+1,j)}), \\ b2(j) &= (3/2-6s){u_k^{n-1}}_{(i,j)} + 3s({u_k^{n-1}}_{(i-1,j)} + {u_k^{n-1}}_{(i+1,j)}), \\ b3(j) &= (1/3-2s){u_k^{n-2}}_{(i,j)} + s({u_k^{n-2}}_{(i-1,j)} + {u_k^{n-2}}_{(i+1,j)}), \end{aligned}$$

end
solve $Au^{n+1}_{(k+1)(i,:)} = \sum_{m=1}^{3}(-1)^{m+1}bm(j)$
end

2.) We apply the SBDF4 method:
Consider the SBDF4-method

$$\begin{aligned} &(\frac{1}{\triangle t})(\frac{25}{12}u^{n+1} - 4u^n + 3u^{n-1} - \frac{4}{3}u^{n-2} + \frac{1}{4}u^{n-3}) \\ &= 4A_1(u^n) - 6A_1(u^{n-1}) + 4A_1(u^{n-2}) - A_1(u^{n-3}) + A_2(u^{n+1}) \end{aligned} \tag{130}$$

after applying this method to the Equations (153-155) instead of implicit trapezoidal rule, we obtain following sub equations: For $i = k$

$$\begin{aligned} &(\frac{25}{12})(u_k^{n+1}) = (4I + 4\triangle tA_1)(u_k^n) - (3I + 6\triangle tA_1)(u_k^{n-1}) + \\ &(\frac{4}{3}I + 4\triangle tA_1)(u_k^{n-2}) - (\frac{1}{4}I + \triangle tA_1)(u_k^{n-3}) + \triangle tA_2(u_{k-1}^{n+1}) \end{aligned} \tag{131}$$

For $i = k + 1$

$$\begin{aligned} &(\frac{25}{12}I - \triangle tA_2)(u_{k+1}^{n+1}) = (4I + 4\triangle tA_1)(u_k^n) - (3I + 6\triangle tA_1)(u_k^{n-1}) + \\ &(\frac{4}{3}I + 4\triangle tA_1)(u_k^{n-2}) - (\frac{1}{4}I + \triangle tA_1)(u_k^{n-3}) \end{aligned} \tag{132}$$

where $k = 1, 3, 5, \ldots$. After inserting to the Equations (157) and (158) into the Equation (131), the following system of equations can be obtained

$$(25/12){u_k^{n+1}}_{(i,j)} = b1(j) - b2(j) + b3(j) - b4(j) + b5(j) \tag{133}$$

where $j = 1, \ldots, N_y$, $b1, b2, b3, b4$ are N_y-dimensional vectors with components

$$b1(j) = (4 - 8s)u^n_{k(i,j)} + 4s(u^n_{k(i-1,j)} + u^n_{k(i+1,j)}) \quad (134)$$

$$b2(j) = (3 - 12s)u^{n-1}_{k\,(i,j)} + 6s(u^{n-1}_{k\,(i-1,j)} + u^{n-1}_{k\,(i+1,j)}) \quad (135)$$

$$b3(j) = (\frac{4}{3} - 8s)u^{n-2}_{k\,(i,j)} + 4s(u^{n-2}_{k\,(i-1,j)} + u^{n-2}_{k\,(i+1,j)}) \quad (136)$$

$$b4(j) = (\frac{1}{4} - 2s)u^{n-3}_{k\,(i,j)} + s(u^{n-3}_{k\,(i-1,j)} + u^{n-3}_{k\,(i+1,j)}) \quad (137)$$

$$b5(i) = -2s(u^{n+1}_{(k-1)(i,j)}) + s(u^{n+1}_{(k-1)(i,j-1)} + u^{n+1}_{(k-1)(i,j+1)}) \quad (138)$$

where $s = \frac{\triangle t}{\triangle x^2}$.

Similarly the Equation (132) can be written as a system of equation as previous

$$(\frac{25}{12} + 2e)u^{n+1}_{(k+1)(i,j)} - e(u^{n+1}_{(k)\,(i,j-1)} + u^{n+1}_{(k)\,(i,j+1)}) = c1(i,j) \quad (139)$$

where $i = 1, \ldots, Nx$, $j = 1, \ldots, Ny$, $e = \frac{\triangle t}{\triangle y^2}$.

For fixed i,the vector

$$u^{n+1}_{(k+1)}(i,:)$$

may be found by solving the tridiagonal system

$$Au^{n+1}_{(k+1)}(i,:) = \sum_{m=1}^{4}(-1)^{m+1}bm(j)$$

.

1.e)

Consider the SBDF5-method

$$\frac{1}{\triangle t}(\frac{137}{60}u^{n+1} - 5u^n + 5u^{n-1} - \frac{10}{3}u^{n-2} + \frac{5}{4}u^{n-3} - \frac{1}{5}u^{n-4} = A_1u^{n-4} - 5A_1u^{n-3} \quad (140)$$
$$+10A_1u^{n-2} - 10A_1u^{n-1} + 5A_1u^n + A_2u^{n+1}$$

After applying this method to the Equations (153-155) instead of implicit trapezoidal rule, we obtain following sub equations for fixed i,

For $i = k$

$$\frac{137}{60}Iu^{n+1}_k = (5I + 5\triangle tA_1)u^n_k - (5I + 10\triangle tA_1)u^{n-1}_k + (\frac{10}{3}I + 10\triangle tA_1)u^{n-2}_k \quad (141)$$
$$-(\frac{5}{4}I + 5\triangle tA_1)u^{n-3}_k + (\frac{1}{5}I + \triangle tA_1)u^{n-4}_k + \triangle tA_2u^{n+1}_{k-1}$$

For $i = k + 1$

$$(\frac{137}{60}I - \triangle tA_2)u_{k+1}^{n+1} = (5I + 5\triangle tA_1)u_k^n - (5I + 10\triangle tA_1)u_k^{n-1} + \quad (142)$$
$$(\frac{10}{3}I + 10\triangle tA_1)u_k^{n-2} - (\frac{5}{4}I + 5\triangle tA_1)u_k^{n-3} + (\frac{1}{5}I + \triangle tA_1)u_k^{n-4}$$

where $k = 1, 3, 5, \ldots$.

After inserting to the Equations (157) and (158) into the Equation (141), the following system of equations can be obtained for fixed i,

$$\frac{137}{60}u_k^{n+1}(i,j) = b1(j) - b2(j) + b3(j) - b4(j) + b5(j) + b6(i) \quad (143)$$

where $j = 1, \ldots, N_y$, $b1, b2, b3, b4$ are N_y-dimensional vectors with components

$$b1(j) = (5I - 10s)u^n_{k\,(i,j)} + 5s(u^n_{k\,(i-1,j)} + u^n_{k\,(i+1,j)}) \quad (144)$$

$$b2(j) = (5I - 20s){u_k^{n-1}}_{(i,j)} + 10s({u_k^{n-1}}_{(i-1,j)} + {u_k^{n-1}}_{(i+1,j)}) \quad (145)$$

$$b3(j) = (\frac{10}{3}I - 20s){u_k^{n-2}}_{(i,j)} + 10s({u_k^{n-2}}_{(i-1,j)} + {u_k^{n-2}}_{(i+1,j)}) \quad (146)$$

$$b4(j) = (\frac{5}{4}I - 10s){u_k^{n-3}}_{(i,j)} + 5s({u_k^{n-3}}_{(i-1,j)} + {u_k^{n-3}}_{(i+1,j)}) \quad (147)$$

$$b4(j) = (\frac{1}{5}I - 2s){u_k^{n-4}}_{(i,j)} + s({u_k^{n-4}}_{(i-1,j)} + {u_k^{n-4}}_{(i+1,j)}) \quad (148)$$

$$b6(i) = -2s(u^{n+1}_{(k-1)(i,j)}) + s(u^{n+1}_{(k-1)(i,j-1)} + u^{n+1}_{(k-1)(i,j+1)}) \quad (149)$$

where $s = \frac{\triangle t}{\triangle x^2}$.

Similarly the Equation (142) can be written as a system of equation as previous

$$(\frac{137}{60}I + 2e)u^{n+1}_{(k+1)(i,j)} - e({u^{n+1}_{(k)}}_{(i,j-1)} + {u^{n+1}_{(k)}}_{(i,j+1)}) = c1(i,j) \quad (150)$$

where $i = 1, \ldots, Nx$, $j = 1, \ldots, Ny$ $e = \frac{\triangle t}{\triangle y^2}$. The vector $c1(i,:)$ is an Nx dimensional with component

For fixed i,the vector

$$u^{n+1}_{(k+1)}(i,:)$$

maybe found by solving the tridiagonal system

$$Au^{n+1}_{(k+1)}(i,:) = \sum_{m=1}^{5}(-1)^{m+1}bm(j)$$

.

6.2 Stability Function II

In this section, we derive the stability function for the operators A_1 and A_2 obtained from finite difference approximation of the unbounded differential operators. In this section we will derive the stability functions depend on both time and space discretization.

As an example, consider an ODEs system

$$\frac{dU}{dt} = (A_1 + A_2)U \ , \tag{151}$$

$$\text{with } U(0) = U^0, \tag{152}$$

where $U(t)$ is an N-dimensional vector,

$$A_1 = \frac{\partial^2}{\partial x^2} \quad \text{and} \quad A_2 = \frac{\partial^2}{\partial y^2}$$

Iterative Splitting method for two dimensional problem is given as follows:

$$\frac{dU_i}{dt} = A_1 U_i + A_2 U_{i-1} \ , \tag{153}$$

$$\text{with } U_i(0) = U^0, \tag{154}$$

$$\frac{dU_{i+1}}{dt} = A_1 U_i + A_2 U_{i+1} \ , \tag{155}$$

$$\text{with } U_{i+1}(0) = U^0, \tag{156}$$

where $i = 1, 3, 5, \ldots$, and $U_0(t) = 0$ (initialization of the scheme). Corresponding to space discretization of 2D heat equation using central differences formula,

$$(A_1 U_k)_{i,j} = \frac{1}{\triangle x^2}[U_{k_{(i-1,j)}} - 2U_{k_{(i,j)}} + U_{k_{(i+1,j)}}], \tag{157}$$

$$(A_2 U_k)_{i,j} = \frac{1}{\triangle y^2}[U_{k_{(i,j-1)}} - 2U_{k_{(i,j)}} + U_{k_{(i,j+1)}}], \tag{158}$$

.

The formulas given in Equation (157) and (158) are not only discretization for second order differential operator. In general, for the given operators A_1 and A_2, the Z_1 and Z_2 are rewritten as

$$Z_1 = \frac{\tau}{\triangle x^k}\tilde{A_1} \text{ and, } Z_2 = \frac{\tau}{\triangle y^k}\tilde{A_2}, \tag{159}$$

where $\tilde{A_1}$ and $\tilde{A_2}$ constant matrix and k is the order of the finite difference approximation of the unbounded differential operators. Thus, if we substitute Equation (159) into previous stability functions, we have both time and space discretization dependent stability functions.

7 Consistency and Accuracy of the Discretization Schemes

7.1 First order iterative splitting with respect to one operator

In this subsection, we consider the Equation (4),

$$\frac{\partial c_i(t)}{\partial t} = -\lambda_1 c_i(t) \ + \ \lambda_2 c_{i-1}(t), \text{ with } c_i(t^n) = c^n, i = 1, 2, \ldots, m \tag{160}$$

after discretization by SBDF1 (BDF1) method, we obtain following finite difference approximation for the Equation (4) on $[0, \tau]$,

$$c_i(\tau) = \chi c^0 + \chi\tau\lambda_2 c_{i-1}(\tau) \quad i = 1, 2, \ldots, m \tag{161}$$

where

$$\chi = \frac{1}{1+\lambda_1\tau} = 1 - (\lambda_1\tau) + (\lambda_1\tau)^2 - O(\tau^3), \quad |\lambda_1\tau| < 1 \tag{162}$$

It can be easily seen from the Equation (162) that χ is the Pade Approximation of the $e^{-\lambda_1\tau}$, that is,

$$e^{-\lambda_1\tau} = \chi + O(\tau^2) \tag{163}$$

7.1.1 First order accuracy via first order approximation

Case1 : Suppose $c_{i-1} = 0$, in this case iterations becomes,
for i=1,

$$c_1(\tau) = \chi c^0 \tag{164}$$

for i=2,

$$c_2(\tau) = \chi\,(1 + \chi\tau\lambda_2)\,c^0 \tag{165}$$

recursively, we have for i=m,

$$\begin{aligned} c_m(\tau) &= \chi(\sum_{j=0}^{m-1}(-1)^j(\chi\tau\lambda_2)^j\,c^0) && (166)\\ &= (\frac{1}{1+\lambda_1\tau})(\frac{1}{1-\chi\lambda_2\tau}), \quad \text{as} \quad m \to \infty && (167)\\ &= (e^{-\lambda_1\tau} + O(\tau^2))(e^{\lambda_2\tau} + O(\tau^2)) && (168)\\ &= e^{(-\lambda_1+\lambda_2)\tau} + O(\tau^2) && (169) \end{aligned}$$

then the order of the method is

$$|c_{exact} - c_{approximation}| = |e^{(-\lambda_1+\lambda_2)\tau} - c_m(\tau)| \leq C_T\,O(\tau^2.$$

Thus we have following conclusion,

Result1 : if $c_{-1} = 0$ then the number of the iteration has to be chosen as $i = k + 1$.

Case2 : Improved initialization by choosing weight to decrease the iteration number

Suppose $c_{i-1} = wc^0$, the iterations becomes,
for i=1,

$$\begin{aligned} c_1(\tau) &= \chi\,(1 + w\tau\lambda_2)\,c^0, && (170)\\ &= (e^{-\lambda_1\tau} + O(\tau^2))(e^{\lambda_2\tau} + O(\tau^2)), \ \text{for} \ w = 1, && (171)\\ &= e^{(-\lambda_1+\lambda_2)\tau} + O(\tau^2)) && (172) \end{aligned}$$

the order of the method is

$$|c_{exact} - c_{approximation}| = |e^{(-\lambda_1+\lambda_2)\tau} - c_m(\tau)| \leq C_T\, O(\tau^2).$$

Thus we have following conclusion,
Result2 : if $c_{-1} \neq 0$ then the number of the iteration has to be chosen as $i = k$ in order to have a same accuracy for iterative splitting as the base method has.

7.1.2 Higher order accuracy via first order approximation

Proposition 7.1 *There exist an optimal number w such that the order of the accuracy of the iterative splitting (4) becomes k after applying the BDF1 to each subequations.*

Proof 7.1 *We will find such w by construction for k=2. We start with the following recursion relation,*

$$c_i(\tau) \;=\; \chi c^0 + \chi\tau\lambda_2 c_{i-1}(\tau), \quad i = 1, 2, \ldots, m \tag{173}$$

assume that $c_{i-1} = wc^0$, by inserting this into Equation (195), we have

$$\begin{aligned}
c_i(\tau) &= \chi\,(1 + w\tau\lambda_2)\,c^0, && (174)\\
&= (e^{-\lambda_1\tau} - r_1 + O(\tau^3))\,(1 + w\tau\lambda_2)\,c^0, \textit{where, } r_1 = \frac{-{\lambda_1}^2}{2}\tau^2 && (175)\\
&= (e^{-\lambda_1\tau} - r_1 + O(\tau^3))\,(1 + (w_0 + \tau w_1)\tau\lambda_2)\,c^0 && (176)\\
&= (e^{-\lambda_1\tau} - r_1 + O(\tau^3))\,(1 + w_0\lambda_2\tau + \lambda_2 w_1\tau^2)\,c^0 && (177)\\
&= e^{-\lambda_1\tau}(1 + w_0\lambda_2\tau + \lambda_2 w_1\tau^2)c^0 - r_1 c^0 + O(\tau^3) && (178)\\
&= e^{-\lambda_1\tau}(1 + w_0\lambda_2\tau + (\lambda_2 w_1 - e^{\lambda_1\tau} r_1)\tau^2)c^0 + O(\tau^3) && (179)\\
&= e^{-\lambda_1\tau}(1 + w_0\lambda_2\tau + (\lambda_2 w_1 - r_1)\tau^2)c^0 + O(\tau^3). && (180)
\end{aligned}$$

Now, by comparing this with exact solution

$$c_{exact} \;=\; e^{-\lambda_1\tau}(1 + \lambda_2\tau + \frac{\lambda_2^2}{2}\tau^2)c^0 + O(\tau^3). \tag{181}$$

we find

$$w_0 = 1, \quad w_1 = \frac{{\lambda_2}^2 - {\lambda_1}^2}{2\lambda_2} \tag{182}$$

By introducing the weight, we recover the third term of the exact solution of the problem thus the order of the method becomes 2, i.e,

$$|c_{exact} - c_{approximation}| = |e^{(-\lambda_1+\lambda_2)\tau} - c_m(\tau)| \leq C_T O(\tau^3).$$

Remark 7.1 *We may continue the procedure to obtain k the order iterative splitting by defining the weight as*

$$w = \sum_{i=0}^{k} w_i \tau^i,$$

thus

$$|c_{exact} - c_{approximation}| = |e^{(-\lambda_1+\lambda_2)\tau} - c_m(\tau)| \leq C_T O(\tau^{k+1}).$$

7.2 Second order iterative splitting with respect to one operator

Proposition 7.2 *The order of the accuracy of the iterative splitting (4) is two after applying the midpoint to each subequations.*

Proof 7.2 *We obtain following finite difference approximation after discretization the Equation (4) by midpoint method on* $[0, \tau]$,

$$c_i(\tau) = \chi_1 c^0 + \chi_2 \tau \lambda_2 (c_{i-1}(0) + c_{i-1}(\tau))(\tau) \quad i = 1, 2, \ldots, m \tag{183}$$

where χ_1 *is defined as follows if* $|\frac{\lambda_1}{2}\tau| < 1$,

$$\chi_1 = \frac{1 - \frac{\lambda_1}{2}\tau}{1 + \frac{\lambda_1}{2}\tau} \tag{184}$$

$$= 1 - (\lambda_1 \tau) + \frac{\lambda_1^2 \tau^2}{2} + O(\tau^3), \tag{185}$$

$$= e^{-\lambda_1 \tau} + O(\tau^3) \tag{186}$$

and Pade Approximation of the $e^{-\lambda_1 \tau}$ *up to the order* $O(\tau^3)$ *and* χ_2 *is defined as follows if* $|\frac{\lambda_1}{2}\tau| < 1$

$$\chi_2 = \frac{\lambda_2 \tau}{1 + \frac{\lambda_1 \tau}{2}} \tag{187}$$

$$= \lambda_2 \tau - \frac{\lambda_1 \lambda_2 \tau^2}{2} + O(\tau^3), \tag{188}$$

assume that $c_{i-1} = 0$, *by inserting this into Equation (195), we have for i= 1,*

$$c_1(\tau) = \chi_1 c^0 \tag{189}$$

$$\tag{190}$$

for i=2,

$$c_2(\tau) = (\chi_1 + \chi_2(1 + \chi_1))c^0 \tag{191}$$

We can easily see that this approximation does not give the exact solution up to the second order, then we need to compute next iteration as follows,

$$c_3(\tau) = (\chi_1 + \chi_2(1 + (\chi_1 + \chi_2(1 + \chi_1))))c^0 \tag{192}$$

$$= \chi_1(1 + {\chi_1}^{-1}\chi_2(1 + (\chi_1 + \chi_2(1 + \chi_1))))c^0 \tag{193}$$

after expanding the terms up to third order we may see that following result

$$c_3(\tau) = e^{(-\lambda_1 + \lambda_2)\tau} + O(\tau^3). \tag{194}$$

In the next theorem, we claim that the order of the iteration is the same as the order of method that applied to each subequation by using the less iterations via introducing weight w.

Proposition 7.3 *There exist an optimal weight* w *so that the number of the iterations can be reducible by keeping the order of the accuracy of the iterative splitting, which is same as the method applying to each subequations.*

Proof 7.3 *We obtain following finite difference approximation after discretization the Equation (4) by midpoint method on* $[0, \tau]$,

$$\begin{aligned} c_i(\tau) &= \chi_1 c^0 + \chi_2 \tau \lambda_2 (c^0 + c_{i-1}(\tau))(\tau), \\ & \quad i = 1, 2, \ldots, m \end{aligned}$$

where χ_1 *is defined as follows if* $|\frac{\lambda_1}{2}\tau| < 1$,

$$\chi_1 = \frac{1 - \frac{\lambda_1}{2}\tau}{1 + \frac{\lambda_1}{2}\tau} \tag{195}$$

$$= 1 - (\lambda_1 \tau) + \frac{\lambda_1^2 \tau^2}{2} + O(\tau^3), \tag{196}$$

$$= e^{-\lambda_1 \tau} + O(\tau^3) \tag{197}$$

and Pade Approximation of the $e^{-\lambda_1 \tau}$ *up to the order* $O(\tau^3)$ *and* χ_2 *is defined as follows if* $|\frac{\lambda_1}{2}\tau| < 1$

$$\chi_2 = \frac{1}{1 + \frac{\lambda_1 \tau}{2}} \tag{198}$$

$$= 1 - \frac{\lambda_1 \tau}{2} + \frac{\lambda_1^2 \tau^2}{4} + O(\tau^3), \tag{199}$$

assume that $c_{i-1} = \tilde{w} c^0, w = (1 + \tilde{w})$, by inserting this into Equation (195), we have

$$c_i(\tau) = \chi_1 c^0 + \chi_2 w \lambda_2 \tau c^0(\tau) \tag{200}$$

$$= (e^{-\lambda_1 \tau}) \left(1 + {\chi_1}^{-1} \chi_2 w \tau \lambda_2\right) c^0 + O(\tau^3) \tag{201}$$

$$\tag{202}$$

now expand ${\chi_1}^{-1}\chi_2$ as

$${\chi_1}^{-1}\chi_2 = \frac{1}{1 - \frac{\lambda_1 \tau}{2}} \tag{203}$$

$$= 1 + \frac{\lambda_1 \tau}{2} + \frac{\lambda_1^2 \tau^2}{8} + O(\tau^3), \tag{204}$$

By inserting this into previous expression and defining $w = w_0 + \tau w_1$, we have

$$c_i(\tau) = e^{-\lambda_1 \tau}(1 + w_0 \lambda_2 \tau + (w_1 \lambda_2 + \frac{\lambda_1}{2} \lambda_2 w_0)\tau^2) c^0 + O(\tau^3) \tag{205}$$

Now, by comparing this with exact solution

$$c_{exact} = e^{-\lambda_1 \tau}(1 + \lambda_2 \tau + \frac{\lambda_2^2}{2}\tau^2) c^0 + O(\tau^3). \tag{206}$$

we find

$$w_0 = 1, \quad w_1 = \frac{\lambda_2 - \lambda_1}{2} \tag{207}$$

By introducing the weight, we reduced the number of the iteration and we have as before

$$|c_{exact} - c_{approximation}| = |e^{(-\lambda_1 + \lambda_2)\tau} - c_m(\tau)| \leq C_T O(\tau^3).$$

8 Application to two dimensional problems

In this section, we apply the proposed methods to following two dimensional problems.

We consider the following heat equation with the initial and boundary conditions:

$$\frac{\partial u}{dt} = D_x\frac{\partial^2 u}{\partial x^2} + D_y\frac{\partial^2 u}{\partial y^2}\,, \tag{208}$$

$$u(x,y,0) = u_{analy}(x,y,0), \quad on \quad \Omega, \tag{209}$$

$$\frac{\partial u(x,y,t)}{\partial n} = 0, \quad on \quad \Omega\times(0,T), \tag{210}$$

$$\tag{211}$$

where $u(x,y)$ is an scalar function, $\omega = [0,1]\times[0,1]$. The analytical solution of the problem is given by

$$u_{analy}(x,y,t) = exp(-(D_x+D_y)\pi^2 t)\cos(\pi x)\cos(\pi y). \tag{212}$$

For approximation error, we choose L_∞ and L_1 which are given by

$$err_{L_\infty} := max(max(|u(x_i,y_j,t^n) - u_{analy}(x_i,y_j,t^n)|))$$

$$err_{L_1} := \sum_{i,j=1}^{m} \triangle x\triangle y|u(x_i,y_j,t^n) - u_{analy}(x_i,y_j,t^n)|$$

$$err_{L_1} := |u(t^n) - u_{analy}(t^n)|$$

Our numerical results obtained by ADI ,SBDF2,SBDF3 and SBDF4 methods are presented in Table 1 , Table 2 and Table 3.

First, we fixed diffusion coefficients as $D_x = D_y = 1$ with the time step dt=0.0005. Comparison of L_∞, L_1 at T=0.5 and CPU time are presented in Table 1 for various spatial step sizes.

Table 1: Comparison of errors at T=0.5 with various Δx and Δy when $D_x = D_y = 1$ and $dt = 0.0005$.

	Δx=Δy	err_{L_∞}	err_{L_1}	CPU times
ADI	1/2	2.8374e-004	2.8374e-004	0.787350
	1/4	3.3299e-005	2.4260e-005	2.074755
	1/16	1.6631e-006	8.0813e-007	22.227760
SBDF2	1/2	2.811e-004	2.811e-004	0.167132
	1/4	3.2519e-005	2.3692e-005	0.339014
	1/16	1.1500e-006	5.5882e-007	2.907924
SBDF3	1/2	2.7841e-004	2.7841e-004	0.312774
	1/4	3.1721e-005	2.3111e-005	0.460088
	1/16	6.2388e-007	3.0315e-007	4.217704
SBDF4	1/2	2.7578e-004	2.7578e-004	0.400968
	1/4	3.0943e-005	2.2544e-005	0.718028
	1/16	1.1155e-007	5.4203e-008	5.550207

In the second experiment, the diffusion coefficients are fixed as $D_x = D_y = 0.001$ for the same time step. Comparison of errors L_∞, L_1 at T=0.5 and CPU time are presented in Table 2 for various spatial steps, Δx and Δy.

Table 2: Comparison of errors at T=0.5 with various Δx and Δy when $D_x = D_y = 0.001$ and $dt = 0.0005$.

	Δx=Δy	err_{L_∞}	err_{L_1}	CPU times
ADI	1/2	0.0019	0.0019	0.786549
	1/4	4.9226e-004	3.5864e-004	2.090480
	1/16	3.1357e-005	1.5237e-005	22.219374
SBDF2	1/2	0.0018	0.0018	0.167021
	1/4	4.8298e-004	3.5187e-004	0.341781
	1/16	2.1616e-005	1.0503e-005	2.868618
SBDF3	1/2	0.0018	0.0018	0.215563
	1/4	4.7369e-004	3.4511e-004	0.461214
	1/16	1.1874e-005	5.7699e-006	4.236695
SBDF4	1/2	0.0018	0.0018	0.274806
	1/4	4.6441e-004	3.3835e-004	0.717014
	1/16	2.1330e-006	1.0365e-006	5.517444

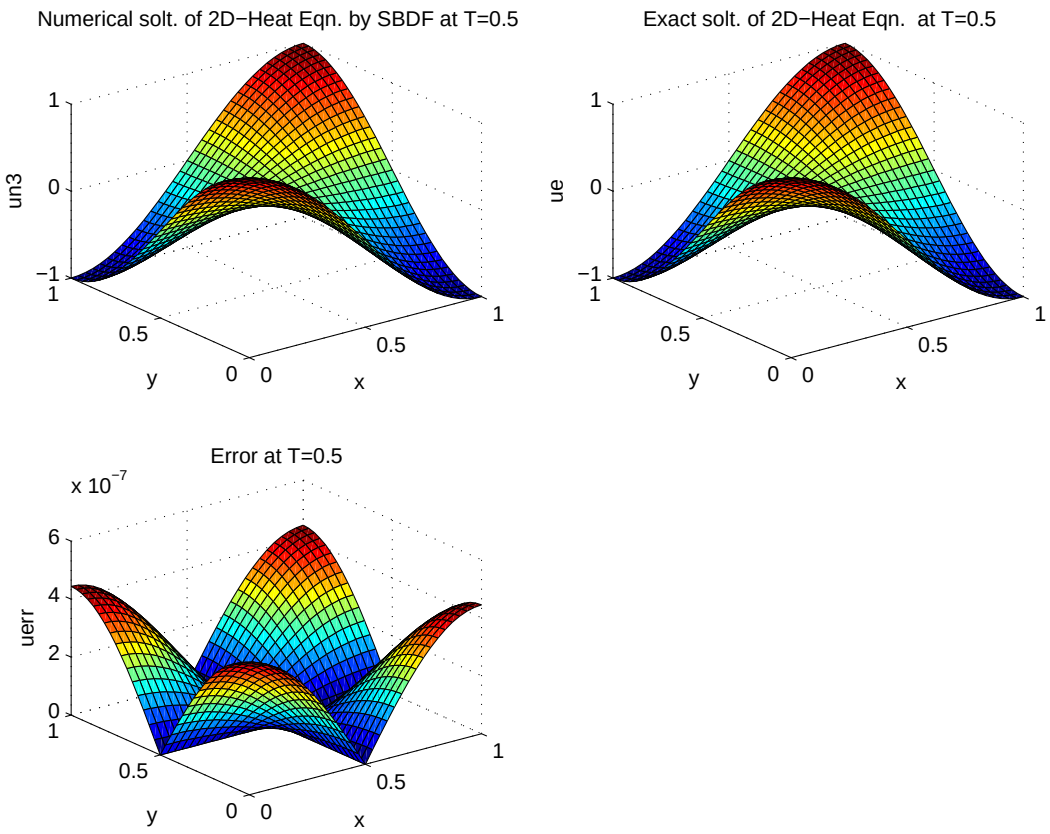
In the third experiment, the diffusion coefficients are fixed as $D_x = D_y = 0.00001$ for the same time step. Comparison of errors L_∞, L_1 at T=0.5 and CPU time are presented in Table 3 for various spatial steps, Δx and Δy.

Table 3: Comparison of errors at T=0.5 with various Δx and Δy when $D_x = D_y = 0.00001$ and $dt = 0.0005$.

	Δx=Δy	err_{L_∞}	err_{L_1}	CPU times
ADI	1/2	1.8694e-005	1.8694e-005	0.783630
	1/4	4.9697e-006	3.6207e-006	2.096761
	1/16	3.1665e-007	1.5386e-007	22.184733
SBDF2	1/2	1.8614e-005	1.8614e-005	0.167349
	1/4	4.8760e-006	3.5524e-006	0.342751
	1/16	2.1828e-007	1.0606e-007	2.864787
SBDF3	1/2	1.8534e-005	1.8534e-005	0.216137
	1/4	4.7823e-006	3.4842e-006	0.465666
	1/16	1.1991e-007	5.8265e-008	4.256818
SBDF4	1/2	1.8454e-005	1.8454e-005	0.399424
	1/4	4.688e-006	3.4159e-006	0.714709
	1/16	2.1539e-008	1.0466e-008	5.501323

Table 4: Comparison of errors at T=0.5 with various Δx and Δy when $D_x = 1$,$D_y = 0.001$ and $dt = 0.0005$.

	Δx=Δy	err_{L_∞}	err_{L_1}	CPU times
ADI	1/2	0.0111	0.0111	0.787006
	1/4	0.0020	0.0015	2.029179
	1/16	1.1426e-004	5.5520e-005	21.959890
SBDF2	1/2	0.0109	0.0109	0.210848
	1/4	0.0019	0.0014	0.385742
	1/16	2.5995e-005	1.2631e-005	2.913781
SBDF3	1/2	0.0108	0.0108	0.316777
	1/4	0.0018	0.0013	0.454392
	1/16	4.4834e-005	2.1785e-005	4.227773
SBDF4	1/2	0.0106	0.0106	0.395751
	1/4	0.0017	0.0013	0.709488
	1/16	1.1445e-004	5.5613e-005	5.562917



Remark 8.1 *In the experiments we obtained improved results by combining the correct time-discretization method with the iterative steps. By using higher order SBDF method we can obtain larger time-steps with sufficient accurate results. Nevertheless higher splitting schemes are expensive and only with large time scales, the benefits are useful.*

9 Conclusion

In the paper we presented the benefits of handling large and small time-scales with iterative operator splitting schemes. So large time-scale can be taken into account with higher order splitting schemes and the computational effort is less. In the stability and consistency analysis we have proven the applicability of the iterative scheme with benefits of weighting.

The applications in parabolic equations shows the verification of the theoretical results. can be done for parabolic equations with nonlinear In future we will discuss the application to time-dependent equations and real-life applications.

References

[1] R.J. Braun B.T Murray J Soto Jr. *Adaptive Finite-difference computations of dendritic growth using a phase-field model.* Modelling Simul. Mater. SC?. eNG. 5, 19957 365-380.

[2] R.E. Ewing. Up-scaling of biological processes and multiphase flow in porous media. *IIMA Volumes in Mathematics and its Applications*, Springer-Verlag, 295 (2002), 195-215.

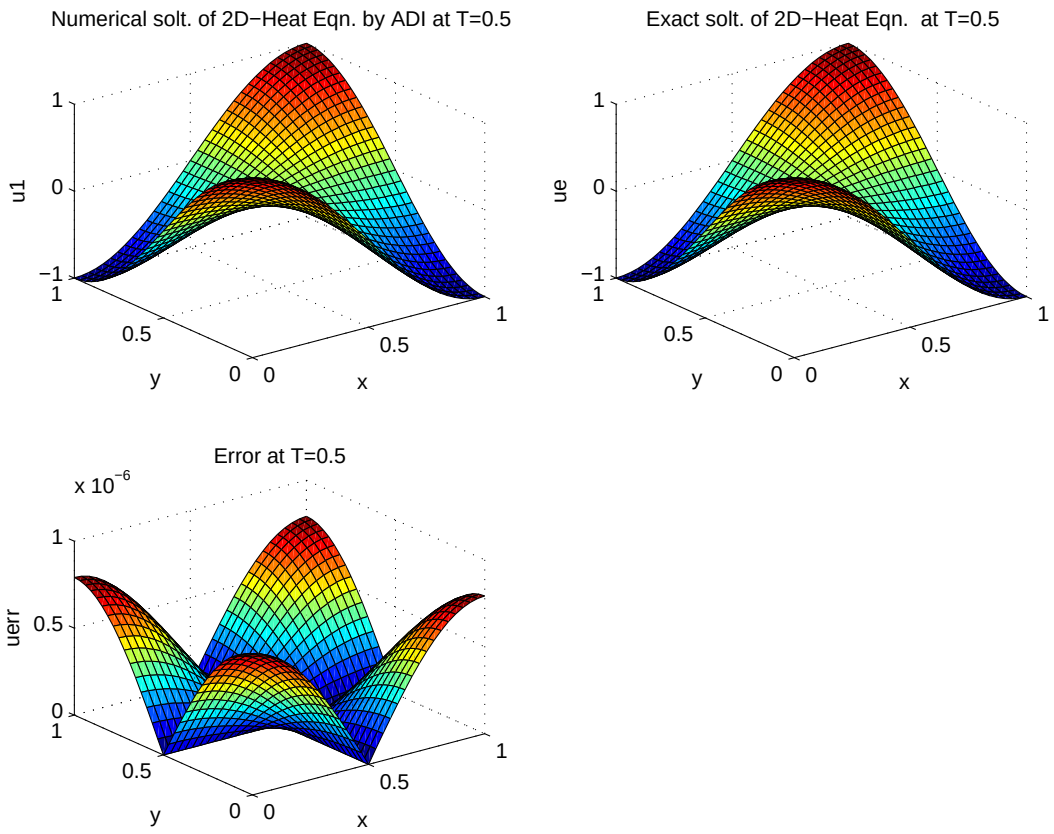


Figure 1: Comparison of errors at T=0.5 with various Δx and Δy for $D_x = D_y = 0.01$ and $dt = 0.0005$.

[3] I. Farago, and Agnes Havasi. *On the convergence and local splitting error of different splitting schemes.* Eötvös Lorand University, Budapest, 2004.

[4] P. Csomós, I. Faragó and A. Havasi. *Weighted sequential splittings and their analysis.* Comput. Math. Appl., (to appear)

[5] K.-J. Engel, R. Nagel, *One-Parameter Semigroups for Linear Evolution Equations.* Springer, New York, 2000.

[6] I. Farago. *Splitting methods for abstract Cauchy problems.* Lect. Notes Comp.Sci. 3401, Springer Verlag, Berlin, 2005, pp. 35-45

[7] I. Farago, J. Geiser. *Iterative Operator-Splitting methods for Linear Problems.* Preprint No. 1043 of the Weierstrass Institute for Applied Analysis and Stochastics, Berlin, Germany, June 2005.

[8] P. Frolkovič and J. Geiser. *Numerical Simulation of Radionuclides Transport in Double Porosity Media with Sorption.* Proceedings of Algorithmy 2000, Conference of Scientific Computing, 28-36, 2000.

[9] J. Geiser. *Numerical Simulation of a Model for Transport and Reaction of Radionuclides.* Proceedings of the Large Scale Scientific Computations of Engineering and Environmental Problems, Sozopol, Bulgaria, 2001.

[10] J. Geiser. *Gekoppelte Diskretisierungsverfahren für Systeme von Konvektions-Dispersions-Diffusions-Reaktionsgleichungen.* Doktor-Arbeit, Universität Heidelberg, 2003.

[11] J. Geiser. *R^3T : Radioactive-Retardation-Reaction-Transport-Program for the Simulation of radioactive waste disposals.* Proceedings: Computing, Communications and Control Technologies: CCCT 2004, The University of Texas at Austin and The International Institute of Informatics and Systemics (IIIS), to appear, 2004.

[12] J. Geiser. *Iterative Operator-Splitting methods for Parabolic Differential Equations : Convergence theory.* Humboldt-Preprint, to be submitted, February 2006.

[13] W. Hackbusch. *Multi-Gird Methods and Applications.* Springer-Verlag, Berlin, Heidelberg, 1985.

[14] W. Hackbusch. *Iterative L"osung gro"ser schwachbesetzter Gleichungssysteme.* Teubner Studienb"ucher: Mathematik, B.G. Teubner Stuttgart, 1993.

[15] W. Hundsdorfer, L. Portero. A Note on Iterated Splitting Schemes. CWI Report MAS-E0404, Amsterdam, Netherlands, 2005.

[16] W.H. Hundsdorfer, J. Verwer W. *Numerical solution of time-dependent advection-diffusion-reaction equations*, Springer, Berlin, (2003).

[17] J. Kanney, C. Miller and C. Kelley. *Convergence of iterative split-operator approaches for approximating nonlinear reactive transport problems.* Advances in Water Resources, 26:247–261, 2003.

[18] R.J. LeVeque. *Finite Volume Methods for Hyperbolic Problems.* Cambridge Texts in Applied Mathematics ,

[19] G.I Marchuk. *Some applicatons of splitting-up methods to the solution of problems in mathematical physics.* Aplikace Matematiky, 1 (1968) 103-132.

[20] G. Strang. *On the construction and comparision of difference schemes.* SIAM J. Numer. Anal., 5:506–517, 1968.

[21] J.,G. Verwer and B. Sportisse. *A note on operator splitting in a stiff linear case.* MAS-R9830, ISSN 1386-3703, 1998.

[22] S. Vandewalle. *Parallel Multigrid Waveform Relaxation for Parabolic Problems.* Teubner Skripten zur Numerik, B.G. Teubner Stuttgart, 1993.

[23] H. Yserentant. *Old and New Convergence Proofs for Multigrid Methods.* Acta Numerica, 285–326, 1993.

[24] Z. Zlatev. *Computer Treatment of Large Air Pollution Models.* Kluwer Academic Publishers, 1995.