A-STABLE HIGH ORDER BLOCK IMPLICIT METHODS FOR PARABOLIC EQUATIONS*

SHISHUN LI†, JING-YUAN WANG‡, AND XIAO-CHUAN CAI§

Abstract. In this paper, we consider the time integration of parabolic equations with block implicit methods (BIM). Depending on the size of the block, high order BIM with A-stability are designed without the need of multiple initial guesses. Similar to Runge-Kutta methods, a BIM can be defined by a tableau including two matrices and two vectors. In addition to the general methodology of BIM, we show a special scheme defined by a positive definite matrix and a positive diagonal matrix; both matrix properties are desirable but not available in Runge-Kutta methods. Moreover, we show that the traditional finite element theory for parabolic problems discretized by the backward Euler or Crank-Nicolson schemes can also be extended to BIM. Finally, we introduce some domain decomposition preconditioners for the linear systems of algebraic equations arising from the block implicit discretization in time and finite element in space. Some numerical results are also reported to show the effectiveness of BIM.

Key words. Block implicit method, A-stability, domain decomposition preconditioner, finite element, parabolic problems.

1. Introduction.

Many time-stepping methods are available for the time integration of parabolic equations [9, 23, 24, 48]. There are two families of classical methods: one-step (multistage) methods and linear multistep (one stage) methods (LMM). Runge-Kutta and Adams methods are examples that are widely used. The methods can be used in their explicit form if the problem is non-stiff or in the implicit form if the problem is stiff. LMM is relatively easy to implement but limited to the Dahlquist order stability barrier [18] and start-up issues [14, 22]. On the other hand, implicit Runge-Kutta methods (IRK) possess favorable stability properties, offer high order of accuracy, and no start-up issues. Among the varieties of IRK, diagonally IRK (DIRK) is preferred by most researchers and is widely used in practice due to its relative ease of implementation [32]. However, DIRK may suffer from the reduction of the order of accuracy and the stage-order is limited to two [24]. The fully IRK (FIRK) methods improve the stage-order, but a higher computational cost is required. Other high order methods with good stability properties have also been studied extensively, such as the general linear methods [9], Taylor Series (multi-derivative) methods [21], boundary value methods [8], exponential time-differencing methods [12, 13, 19, 26]. One particular class of methods that deserves further study and has the potential for many important applications is the so-called block implicit methods (BIM), which was developed by coupling multiple classical LMM schemes in a single method [7, 44, 45, 51, 52]. BIM overcomes the Dahlquist stability barriers and the start-up issues of LMM.

The idea of BIM was first proposed in [41], where the methods were used to provide starting values for predictor-corrector schemes. A BIM with block size k can be described by a tableau

$$\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{a}^T & \boldsymbol{b}^T \end{array},$$

*The work was partly supported by NSFC-RGC 62161160312, NSF of Henan 212300410347, and FDCT 0141/2020/A3, 0079/2021/AFJ.

†School of Mathematics and Information Science, Henan Polytechnic University, Jiaozuo 454003, P.R. China (lss6@sina.com).

‡Department of Mathematics, University of Macau, Macau SAR, China (yc27479@um.edu.mo).

§Department of Mathematics, University of Macau, Macau SAR, China (xccai@um.edu.mo).

where $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{k\times k}$ and $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^k$ can be derived from the coefficients of LMM, and in this paper we refer to them as the *BIM matrices* and *BIM vectors*. BIM circumvents the order and stability barrier of LMM, and share some of the nice features of IRK [2, 5–7, 42]. However, BIM is not widely used because it requires the solving of a large and ill-conditioned algebraic system of equations at each iteration. For example, the discretization of a linear parabolic equation by BIM in time and the finite element in space produces a system of equations of the form

$$\boldsymbol{A} \otimes \boldsymbol{M} + \tau \boldsymbol{B} \otimes \boldsymbol{K}, \tag{1.1}$$

where $\otimes$ denotes the Kronecker tensor product, τ is the time step size, $\boldsymbol{M}, \boldsymbol{K} \in \mathbb{R}^{N\times N}$ denote the mass and the stiffness matrix, respectively. N is the number of unknowns in space. (1.1) is larger and more difficult to deal with than the coefficients matrices resulting from, for examples, the implicit Euler, Crank-Nicolson and LMM that are of the form $\alpha\boldsymbol{M} + \beta\tau\boldsymbol{K}$, where $\alpha > 0$ and $\beta > 0$ are constants.

One of the goals of this paper is to continue the study of BIM initiated in [51, 52]. We find the explicit relation between the BIM matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ by employing the order conditions of LMM. This is important for the theoretical analysis of BIM and helpful for the construction of new algorithms with desirable properties. By employing the techniques in [3, 4, 27], we construct a special type of BIM algorithms such that the matrix $\boldsymbol{B}^{-1}\boldsymbol{A}$ is diagonally stable. In these algorithms, $\boldsymbol{B}$ is a positive diagonal matrix and $\boldsymbol{A}$ is a positive definite matrix. As a result, the matrix (1.1) has the same positive definiteness as the matrix $\alpha\boldsymbol{M} + \beta\tau\boldsymbol{K}$, and the corresponding linear systems are easier to precondition and solve. On the other hand, the FIRK matrices are dense and positive stable; i.e., the real part of the eigenvalues are positive [31, 34, 39], which makes the positive definiteness of the matrix $\boldsymbol{K}$ in (1.1) useless.

In this paper, following the idea of [10, 11, 15, 36–38], we also introduce some parallel DD preconditioners for parabolic partial differential equations (PDEs) discretized by BIM in time and finite element in space. Similar to the implicit Euler and Crank-Nicolson methods for parabolic PDEs [48], the uniqueness of the solution can be proved by the Lax-Milgram theorem, and the a priori error estimates for finite element methods can also be established. Note that such analysis can not be obtained for FIRK since the FIRK matrices are only positive stable but not positive definite [40].

The rest of the paper is organized as follows. In Section 2, we present some motivating examples. In Section 3, we first study BIM with A-stability and present the matrix $\boldsymbol{B}^{-1}\boldsymbol{A}$ explicitly, then we construct BIM by selecting special $\boldsymbol{A}$ and $\boldsymbol{B}$. In Section 4, a comparison with FIRK is presented to show that BIM is more competitive. Some DD preconditioners in the tensor form are introduced in Section 5 for linear parabolic PDEs discretized with BIM in time and finite element in space. Finally, some conclusions are given in Section 6.

2. Block implicit methods: Formulation, order of convergence and stability. In this section, we first review briefly the one-step and multistep algorithms. Then present some motivating examples to show why BIM is important. In particular we show how to combine a few "not-so-good"methods (in terms of order, and/or stability, and/or the requirement of multiple initial values) into a single block method with all the desirable properties. After these interesting examples, we derive the general form of BIM and introduce several types of stabilities.

2.1. A brief review of time-stepping methods. We consider a one-dimensional initial value problem

$$\left\{\begin{array}{ll} y' &= f(t,y), \quad t \in (0,T], \\ y(t_0) &= y_0 \end{array}\right. \tag{2.1}$$

discretized on a uniform temporal mesh $0 = t_0 < t_1 < \cdots < t_N = T$ with the time step size $\tau = t_n - t_{n-1}$. We denote by y_n as the solution at time t_n. A one-step method for solving (2.1) can be described as

$$y_{n+1} = R(z)y_n, \; n = 0, 1, \ldots, N-1, \tag{2.2}$$

where $R(z) = P(z)/Q(z)$ is a rational function, $P(z)$ and $Q(z)$ are polynomials of degree m and j, respectively. The function $R(z)$ is called the stability function of the method [24], and it can be interpreted as the numerical solution after one step of the Dahlquist test equation

$$y' = \lambda y, \quad y_0 = 1, \quad z = \tau\lambda. \tag{2.3}$$

The set $S = \{z \in \mathbb{C} : \; |R(z)| \leq 1\}$ is called the stability domain of the method.

Definition 2.1. *(Dahlquist 1963 [18]) A method, whose stability domain satisfies*

$$S \supset \mathbb{C}^- = \{z : \; \text{Re } z < 0\},$$

is called A-stable, where $\mathbb{C}^-$ denotes the entire left half-plane.

Suppose $R(z)$ is an arbitrary rational approximation of order p with m zeros and j poles. The following theorems for the A-stability of $R(z)$ can be found in [24].

Theorem 2.2. *(Crouzeix & Ruamps 1977 [17]) Suppose $p \geq 2j-2$, $|R(\infty)| \leq 1$, and the coefficients of the denominator $Q(z)$ have alternating signs. Then, $R(z)$ is A-stable.*

Theorem 2.3. *Suppose $p \geq 2j-3$, $R(z)$ is I-stable, and the coefficients of $Q(z)$ have alternating signs. Then, $R(z)$ is A-stable.*

A k-step LMM is often written as

$$\alpha_k y_{n+k} + \cdots + \alpha_1 y_{n+1} + \alpha_0 y_n = \tau(\beta_k f_{n+k} + \cdots + \beta_1 f_{n+1} + \beta_0 f_n), \tag{2.4}$$

where $\alpha_k = 1$, $|\alpha_0| + |\beta_0| > 0$. If (2.4) is of order p, the coefficients α_j and β_j ($j = 0, 1, \cdots, k$) satisfy

$$\left\{\begin{array}{ll} c_0 &= \sum_{j=0}^{k} \alpha_j = 0, \\ c_1 &= \sum_{j=0}^{k} j\alpha_j - \sum_{j=0}^{k} \beta_j = 0, \\ c_2 &= \frac{1}{2}\sum_{j=0}^{k} j^2\alpha_j - \sum_{j=0}^{k} j\beta_j = 0, \\ \cdots\cdots & \\ c_p &= \frac{1}{p!}\sum_{j=0}^{k} j^p\alpha_j - \frac{1}{(p-1)!}\sum_{j=0}^{k} j^{p-1}\beta_j = 0, \qquad p = 3, 4, \cdots. \end{array}\right. \tag{2.5}$$

(2.5) is called the order conditions. Although LMM is relatively easy to implement, it is limited by the Dahlquist order stability barrier [18].

2.2. Some motivating examples. We consider two simple equations.

EXAMPLE 1. $y' = -3y, \quad y(0) = 1, \quad t \in (0, 2]$.

EXAMPLE 2. $y' = -2000(y - \cos(t)), \quad y(0) = 0, \quad t \in (0, 1.5]$.

Below we mention three methods to serve as the basis of the discussion.

Method A (second-order):

$$y_{n+2} - y_n = 2\tau f_{n+1}, \quad \text{with given } y_0 \text{ and } y_1, \tag{2.6}$$

which is a two-step method often referred to as the mid-point rule.

Method B (third-order):

$$y_{n+2} + 4y_{n+1} - 5y_n = \tau(4f_{n+1} + 2f_n), \quad \text{with given } y_0 \text{ and } y_1, \tag{2.7}$$

which is a two-step explicit method.

Method C (third-order):

$$y_{n+2} - 3y_{n+1} + 2y_n = \frac{\tau}{12}(7f_{n+2} - 8f_{n+1} - 11f_n), \quad \text{with given } y_0 \text{ and } y_1, \tag{2.8}$$

which is a two-step implicit method.

Methods A, B and C are two-step linear methods and they are known to be not stable, and in Figure 1, the instability can be seen clearly from the numerical solutions obtained with several different mesh sizes and a comparison with the known exact solution of Example 1. Because of the instability, these methods are never used in practice. Next, we come up with some block implicit methods designed by combining some stable and unstable methods into a system of methods. Methods D, E and F shown below are designed by coupling two or three classical linear multistep methods (stable or unstable), but only one given initial value y_0. Since they are block methods, the solutions at two or three time steps are obtained simultaneously. It is interesting to note that these three methods are all stable from Figure 2. Moreover, Table 1 shows that Methods D, E and F are second-order, fourth-order and third-order, respectively.

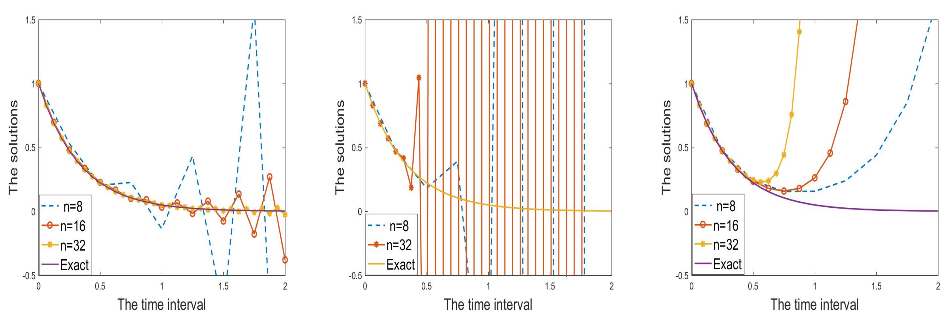

Fig. 1: Numerical solutions of Example 1 computed by the Methods A (left), B (middle) and C (right)

Method D (second-order):

$$\begin{cases} y_{n+2} - y_n &= 2\tau f_{n+1}, \\ 3y_{n+2} - 4y_{n+1} + y_n &= 2\tau f_{n+2}, \end{cases} \tag{2.9}$$

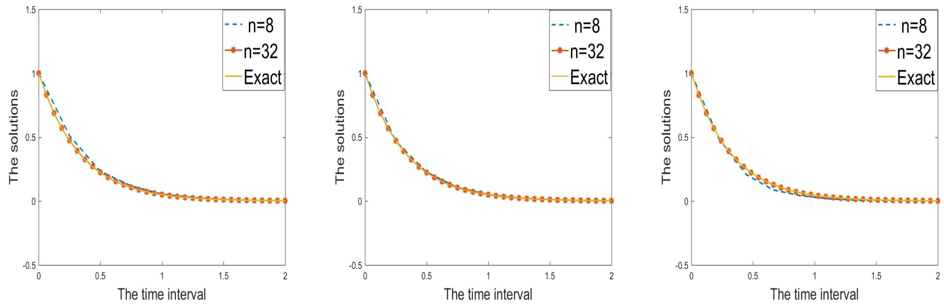

Fig. 2: Numerical solutions of Example 1 computed by the Methods D (left), E (middle) and F (right)

Table 1: The order and the error $|y(t) - y|_\infty$ of Methods D, E and F for solving Example 1.

n	8	16	32	64	128	256	512	1024
Method D	3.92e-2	9.95e-3	2.20e-3	5.42e-4	1.34e-4	3.36e-5	8.42e-6	2.10e-6
		1.97	2.17	2.02	2.01	1.99	1.99	2.00
Method E	4.62e-3	4.76e-4	3.88e-5	2.79e-6	1.87e-7	1.21e-8	7.71e-10	4.86e-11
		3.27	3.61	3.79	3.89	3.95	3.97	3.98
Method F	1.91e-2	3.32e-3	3.89e-4	4.22e-5	5.05e-6	6.13e-7	7.53e-8	9.33e-9
		2.52	3.09	3.20	3.06	3.04	3.02	3.01

which is an one-step implicit method (with initial value $y_0 = y(0)$) obtained by coupling (2.6) and the BDF2 rule.

Method E (fourth-order):

$$\begin{cases} y_{n+2} + 4y_{n+1} - 5y_n &= \tau(4f_{n+1} + 2f_n), \\ y_{n+2} - 3y_{n+1} + 2y_n &= \frac{\tau}{12}(7f_{n+2} - 8f_{n+1} - 11f_n), \end{cases} \tag{2.10}$$

which is an one-step implicit method (with initial value $y_0 = y(0)$) obtained by coupling (2.7) and (2.8).

Method F (third-order):

$$\begin{cases} 12y_{n+3} - 12y_{n+2} &= \tau(23f_{n+2} - 16f_{n+1} + 5f_n), \\ 11y_{n+3} - 18y_{n+2} + 9y_{n+1} - 2y_n &= 6\tau f_{n+3}, \\ 3y_{n+3} - 3y_{n+1} &= \tau(-2f_{n+3} + 13f_{n+2} - 8f_{n+1} + 3f_n), \end{cases} \tag{2.11}$$

which is an one-step implicit method (with initial value $y_0 = y(0)$) obtained by coupling three linear 3-step methods.

It is clear that the new block methods D, E and F all have better stability than the classical linear multistep method. The solutions at k time steps may behave differently since the corresponding stability functions have different properties. Below we introduce some terminologies to describe these stabilities:

Definition 2.4. *Assume that $y_{n+1}, y_{n+2}, \cdots, y_{n+k}$ are computed simultaneously by employing the initial value y_n. Then, the method is*

(1) strong k-step stable if $|y_{n+k}| \le |y_{n+k-1}| \le \cdots \le |y_{n+1}| \le |y_n|$;

(2) k-step stable if $|y_{n+i}| \le |y_n|$ for all $1 \le i \le k$;
(3) k^{th}-step stable if $|y_{n+k}| \le |y_n|$.

Note that this definition is exactly the same as that for the classical one-step or multistep methods when $k = 1$.

2.3. Derivation of block implicit methods. In this section, we derive a family of block implicit methods based on the classical linear multistep methods.

DEFINITION 2.5. *(Block implicit method with block size k) For a given initial value y_n, the problem (2.1) is solved by the following k by k system of algebraic equations*

$$
\begin{aligned}
&(2.12.1) \quad a_{1k}y_{n+k} + \cdots + a_{11}y_{n+1} + a_{10}y_n = \tau(b_{1k}f_{n+k} + \cdots + b_{11}f_{n+1} + b_{10}f_n),\\
&(2.12.2) \quad a_{2k}y_{n+k} + \cdots + a_{21}y_{n+1} + a_{20}y_n = \tau(b_{2k}f_{n+k} + \cdots + b_{21}f_{n+1} + b_{20}f_n),\\
&\qquad\qquad\vdots\\
&(2.12.\mathrm{k}) \quad a_{kk}y_{n+k} + \cdots + a_{k1}y_{n+1} + a_{k0}y_n = \tau(b_{kk}f_{n+k} + \cdots + b_{k1}f_{n+1} + b_{k0}f_n),
\end{aligned}
$$

where $|a_{i0}| + |b_{i0}| > 0$ and each formula (2.12.i) $(1 \le i \le k)$ satisfies the order conditions (2.5). The error is $\boldsymbol{C}_{p+1} = (c_{1,p+1}\ c_{2,p+1}\ \cdots\ c_{k,p+1})^T$ and $c_{i,p+1}$ is the error constant of the formula (2.12.i) $(1 \le i \le k)$.

It is convenient to represent a BIM by a partitioned tableau of the form

$$\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{a}^T & \boldsymbol{b}^T \end{array},$$

where $\boldsymbol{a} = (a_{10}\ a_{20}\ \cdots\ a_{k0})^T$, $\boldsymbol{b} = (b_{10}\ b_{20}\ \cdots\ b_{k0})^T$, and the BIM matrices

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{pmatrix} \quad \text{and} \quad \boldsymbol{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1k} \\ b_{21} & b_{22} & \cdots & b_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ b_{k1} & b_{k2} & \cdots & b_{kk} \end{pmatrix}.$$

Applying the above BIM to

$$y' = 0,$$

we have

$$\boldsymbol{A}\boldsymbol{y} = -\boldsymbol{a}y_n, \tag{2.13}$$

where $\boldsymbol{y} = (y_{n+1}\ y_{n+2}\ \cdots\ y_{n+k})^T$. Since $a_{ik} + \cdots + a_{i1} + a_{i0} = 0$ (consistency condition of LMM), it follows from Cramer's rule that

$$y_{n+i} = \frac{|\boldsymbol{A}_i|}{|\boldsymbol{A}|}y_n = y_n, \quad (1 \le i \le k),$$

where $\boldsymbol{A}_i$ is the matrix $\boldsymbol{A}$ with the i^{th} column $\boldsymbol{a}_i$ replaced by the vector $-\boldsymbol{a}$. Therefore, the method is stable if $\boldsymbol{A}$ is nonsingular. Now, we introduce the definition for the stability of BIM.

DEFINITION 2.6. *(Block zero-stability) The BIM (2.12) is called k-step stable, if the BIM matrix $\boldsymbol{A}$ is nonsingular.*

Applying the above BIM to *Dahlquist* test equation (2.3), we have

$$\boldsymbol{C}(z)\boldsymbol{y} \equiv (\boldsymbol{A} - z\boldsymbol{B})\boldsymbol{y} = (z\boldsymbol{b} - \boldsymbol{a})y_n, \tag{2.14}$$

where $z = \lambda\tau$.

Define the rational functions

$$R_{n+i}(z) = \frac{P_{n+i}(z)}{Q_{n+i}(z)}, \quad 1 \le i \le k. \tag{2.15}$$

Here $P_{n+i}(z) = |\boldsymbol{C}_i(z)| = |\boldsymbol{A}_i - z\boldsymbol{B}_i|$ and $Q_{n+i}(z) = |\boldsymbol{C}_{i-1}(z)|$ or $Q_{n+i}(z) = |\boldsymbol{C}(z)|$, where $\boldsymbol{C}_0(z) = \boldsymbol{C}(z)$ and $\boldsymbol{B}_i$ $(1 \le i \le k)$ is the matrix obtained by replacing column i of $\boldsymbol{B}$ by $-\boldsymbol{b}$. Note that if (2.12) is block k-step stable, i.e., $\boldsymbol{A}$ is nonsingular, the polynomial $|\boldsymbol{C}_i(z)|$ is nonzero since the constant term is $|\boldsymbol{A}|$. From (2.14), we have $y_{n+i} = \frac{|\boldsymbol{C}_i(z)|}{|\boldsymbol{C}(z)|} y_n$ and $y_{n+i} = \frac{|\boldsymbol{C}_i(z)|}{|\boldsymbol{C}_{i-1}(z)|} y_{n+i-1}$.

DEFINITION 2.7. *The sets*

$$S_i = \{z \in \mathbb{C} : \ |R_{n+i}(z)| \le 1\}, \quad 1 \le i \le k$$

are called the stability domains of the BIM.

i) *If $S_i \supset \mathbb{C}^-$ for all $1 \le i \le k$, i.e., each $R_{n+i}(z) = |\boldsymbol{C}_i(z)|/|\boldsymbol{C}_{i-1}(z)|$ is A-stable, this method is called strong block k-step A-stable;*

ii) *If $S_i \supset \mathbb{C}^-$ for all $1 \le i \le k$, i.e., each $R_{n+i}(z) = |\boldsymbol{C}_i(z)|/|\boldsymbol{C}(z)|$ is A-stable, this method is called block k-step A-stable;*

iii) *If $S_k \supset \mathbb{C}^-$, i.e., $R_{n+k}(z) = |\boldsymbol{C}_k(z)|/|\boldsymbol{C}(z)|$ is A-stable, this method is called block k^{th}-step A-stable;*

3. Block implicit methods with A-stability. BIM with A-stability was first studied based on the interpolatory formulas of Newton-Cotes type [51]. The coefficients of $|\boldsymbol{C}(z)|$ and $|\boldsymbol{C}_i(z)|$ can be obtained explicitly by using Newton-Cotes formulas. By using Routh echeme [16], it is proved numerically that the block size of A-stable BIM has to be less than or equal to 8. In this section, we further study BIM with A-stability by different techniques. Based on the Lyapunov stability theorem, we explicitly construct a type of BIM by choosing the BIM matrices $\boldsymbol{A}$ and $\boldsymbol{B}$.

3.1. A-stability of BIM. We first show that the equation (2.14) has a unique solution by employing the order conditions (2.5); i.e., the solution is uniquely determined by $\boldsymbol{B}^{-1}\boldsymbol{A}$, and is independent of the individual BIM matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. Then, we present several BIM for some k and prove their stability. Let

$$\boldsymbol{W} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 2 & \cdots & k \\ 0 & 1 & 2^2 & \cdots & k^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & 2^k & \cdots & k^k \end{pmatrix} \quad \text{and} \quad \boldsymbol{H} = \begin{pmatrix} 0 & & & \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & k & 0 \end{pmatrix}$$

we have the following important lemma in [1].

LEMMA 3.1. *The elements of the matrix $\boldsymbol{W}^{-1}\boldsymbol{H}\boldsymbol{W}$ are*

$$(\boldsymbol{W}^{-1}\boldsymbol{H}\boldsymbol{W})_{ij} = \begin{cases} \displaystyle\sum_{p=1}^{i} \frac{1}{p} - \sum_{p=1}^{k-i} \frac{1}{p}, & i = j, \\ \displaystyle\frac{(-1)^{j-i}}{j-i} \frac{C_i^k}{C_j^k}, & i \ne j, \quad i, j = 0, 1, \ldots, k. \end{cases}$$

Here $C_i^k = \begin{pmatrix} k \\ i \end{pmatrix}$ *denotes the binomial coefficient.*

Let $\boldsymbol{e}_i$ be the column i of the identity matrix $\boldsymbol{I} \in \mathbb{R}^{k\times k}$, $\boldsymbol{e} = (1, 1, \cdots, 1)^T \in \mathbb{R}^k$, $\boldsymbol{e}_0 = (0, 0, \cdots, 0)^T \in \mathbb{R}^k$, $\widetilde{\boldsymbol{x}} = (1, 2, \cdots, k)^T \in \mathbb{R}^k$ and $\boldsymbol{v} = (1, 2^{k+1}, \cdots, k^{k+1})^T$. Define $\widetilde{\boldsymbol{b}} \in \mathbb{R}^k$ and $\boldsymbol{N} \in \mathbb{R}^{k\times k}$ by

$$(-\widetilde{\boldsymbol{b}})_i = \frac{(-1)^i}{C_i^k} \quad \text{and} \quad \boldsymbol{N}_{ij} = \begin{cases} \displaystyle\sum_{p=1}^{i} \frac{1}{p} - \sum_{p=1}^{k-i} \frac{1}{p} + \frac{1}{i}, & i = j, \\ \displaystyle\frac{(-1)^{i-j}}{i-j} \frac{i}{j} \frac{C_j^k}{C_i^k}, & i \neq j. \end{cases} \tag{3.1}$$

By replacing $\boldsymbol{e}_i$ $(i = 1, 2, \ldots, k)$ in the identity matrix $\boldsymbol{I}$ by $\boldsymbol{e}$ and $-\widetilde{\boldsymbol{b}}$, define the matrices $\boldsymbol{Q}_i, \boldsymbol{I}_i \in \mathbb{R}^{k\times k}$ as follows:

$$\boldsymbol{Q}_i = [\boldsymbol{e}_1 \ \cdots \ \boldsymbol{e}_{i-1} \ \boldsymbol{e} \ \boldsymbol{e}_{i+1} \ \cdots \ \boldsymbol{e}_k], \quad \boldsymbol{I}_i = [\boldsymbol{e}_1 \ \cdots \ \boldsymbol{e}_{i-1} \ -\widetilde{\boldsymbol{b}} \ \boldsymbol{e}_{i+1} \ \cdots \ \boldsymbol{e}_k]. \tag{3.2}$$

We obtain the following theorem.

THEOREM 3.2. *For BIM, the solution of (2.14) is unique, and the stability function is*

$$R_{n+i}(z) = \frac{|\boldsymbol{C}_i(z)|}{|\boldsymbol{C}(z)|} = \frac{|\boldsymbol{N}\boldsymbol{Q}_i - z\boldsymbol{I}_i|}{|\boldsymbol{N} - z\boldsymbol{I}|}.$$

The error $\boldsymbol{C}_{p+1} = (\boldsymbol{D}_0\boldsymbol{v} - (p+1)\boldsymbol{N}^{-1}\boldsymbol{v})/(k+2)!$, *where* $\boldsymbol{D}_0 = diag(1, 2, \cdots, k)$.

Proof. Combining the order conditions (2.5) and (2.14), setting $c_2 = c_3 = \cdots = c_{k+1} = 0$, we have

$$\boldsymbol{A}\boldsymbol{D}_0\boldsymbol{C}_1 = \boldsymbol{B}\boldsymbol{C}_1\boldsymbol{D}_1, \tag{3.3}$$

where

$$\boldsymbol{D}_0 = \begin{pmatrix} 1 & & & \\ & 2 & & \\ & & \ddots & \\ & & & k \end{pmatrix}, \ \boldsymbol{D}_1 = \begin{pmatrix} 2 & & & \\ & 3 & & \\ & & \ddots & \\ & & & k+1 \end{pmatrix}, \ \boldsymbol{C}_1 = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 2 & 2^2 & \cdots & 2^k \\ \vdots & \vdots & \ddots & \vdots \\ k & k^2 & \cdots & k^k \end{pmatrix}.$$

Clearly, $\boldsymbol{D}_0$, $\boldsymbol{D}_1$ and $\boldsymbol{C}_1$ are nonsingular. Then, we have

$$\boldsymbol{B} = \boldsymbol{A}\boldsymbol{N}^{-1}, \qquad \boldsymbol{N} = \boldsymbol{C}_1\boldsymbol{D}_1\boldsymbol{C}_1^{-1}\boldsymbol{D}_0^{-1}. \tag{3.4}$$

Therefore,

$$\boldsymbol{C}(z) = (\boldsymbol{A} - z\boldsymbol{B}) = \boldsymbol{A}\boldsymbol{N}^{-1}(\boldsymbol{N} - z\boldsymbol{I}). \tag{3.5}$$

Setting $c_1 = 0$ in (2.5), we have

$$\begin{cases} a_{11} + 2a_{12} + \cdots + ka_{1k} = b_{10} + b_{11} + \cdots + b_{1k}, \\ a_{21} + 2a_{22} + \cdots + ka_{2k} = b_{20} + b_{21} + \cdots + b_{2k}, \\ \qquad \vdots \\ a_{k1} + 2a_{k2} + \cdots + ka_{kk} = b_{k0} + b_{k1} + \cdots + b_{kk}. \end{cases} \tag{3.6}$$

Define $\widetilde{\boldsymbol{x}} = (1, 2, \cdots, k)^T$, (3.6) implies

$$\boldsymbol{A}\widetilde{\boldsymbol{x}} = \boldsymbol{b} + \boldsymbol{B}\boldsymbol{e}.$$

Hence,

$$\boldsymbol{b} = \boldsymbol{A}\widetilde{\boldsymbol{x}} - \boldsymbol{B}\boldsymbol{e} = \boldsymbol{A}\boldsymbol{N}^{-1}(\boldsymbol{N}\widetilde{\boldsymbol{x}} - \boldsymbol{e}). \tag{3.7}$$

Set $\widetilde{\boldsymbol{b}} = \boldsymbol{N}\widetilde{\boldsymbol{x}} - \boldsymbol{e}$, it follows from (3.4) and (3.7) that

$$\widetilde{\boldsymbol{b}} = \boldsymbol{C}_1\boldsymbol{D}_1^{-1}\boldsymbol{C}_1\boldsymbol{D}_0^{-1}\widetilde{\boldsymbol{x}} - \boldsymbol{e} = \boldsymbol{C}_1\boldsymbol{D}_1^{-1}\boldsymbol{C}_1\boldsymbol{e} - \boldsymbol{e} = \boldsymbol{C}_1\boldsymbol{D}_0^{-1}\boldsymbol{C}_1\boldsymbol{e}. \tag{3.8}$$

Setting $c_0 = 0$ in (2.5), we have

$$\boldsymbol{a} = -\boldsymbol{A}\boldsymbol{e}. \tag{3.9}$$

Therefore, combining (3.7), (3.8) and (3.9) we have

$$\begin{aligned}
\boldsymbol{C}_i(z) &= \boldsymbol{A}_i - z\boldsymbol{B}_i \\
&= [\boldsymbol{a}_1 \ \cdots \ \boldsymbol{a}_{i-1} \ -\boldsymbol{a} \ \boldsymbol{a}_{i+1} \ \cdots \ \boldsymbol{a}_k] - z[\boldsymbol{b}_1 \ \cdots \ \boldsymbol{b}_{i-1} \ -\boldsymbol{b} \ \boldsymbol{b}_{i+1} \ \cdots \ \boldsymbol{b}_k] \\
&= \boldsymbol{A}[\boldsymbol{e}_1 \ \cdots \ \boldsymbol{e}_{i-1} \ \boldsymbol{e} \ \boldsymbol{e}_{i+1} \ \cdots \ \boldsymbol{e}_k] - z\boldsymbol{B}[\boldsymbol{e}_1 \ \cdots \ \boldsymbol{e}_{i-1} \ -\widetilde{\boldsymbol{b}} \ \boldsymbol{e}_{i+1} \ \cdots \ \boldsymbol{e}_k] \\
&= \boldsymbol{A}\boldsymbol{N}^{-1}(\boldsymbol{N}\boldsymbol{Q}_i - z\boldsymbol{I}_i). \qquad (3.10)
\end{aligned}$$

It follows from (3.5) and (3.10) that

$$R_{n+i}(z) = \frac{|\boldsymbol{C}_i(z)|}{|\boldsymbol{C}(z)|} = \frac{|\boldsymbol{N}\boldsymbol{Q}_i - z\boldsymbol{I}_i|}{|\boldsymbol{N} - z\boldsymbol{I}|}.$$

Set

$$\boldsymbol{F} = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & 0 \end{pmatrix}_{(k+1)\times(k+1)}.$$

It is clear that

$$\boldsymbol{W}^T\boldsymbol{F} = \begin{pmatrix} 0 & \boldsymbol{e}_1^T \\ \boldsymbol{e}_0 & \boldsymbol{D}_0^{-1}\boldsymbol{C}_1 \end{pmatrix} \quad \text{and} \quad \boldsymbol{H}^T = \boldsymbol{F}\begin{pmatrix} 0 & \boldsymbol{e}_0^T \\ \boldsymbol{e}_0 & \boldsymbol{D}_0 \end{pmatrix}. \tag{3.11}$$

Therefore,

$$\begin{aligned}
\boldsymbol{W}^T\boldsymbol{H}^T\boldsymbol{W}^{-T} &= \boldsymbol{W}^T\boldsymbol{F}\begin{pmatrix} 0 & \boldsymbol{e}_0^T \\ \boldsymbol{e}_0 & \boldsymbol{D}_0 \end{pmatrix}\boldsymbol{W}^{-T} = \begin{pmatrix} 1 & \boldsymbol{e}_0^T \\ \boldsymbol{e} & \boldsymbol{C}_1 \end{pmatrix}\boldsymbol{F}\begin{pmatrix} 0 & \boldsymbol{e}_0^T \\ \boldsymbol{e}_0 & \boldsymbol{D}_0 \end{pmatrix}\boldsymbol{W}^{-T} \\
&= \begin{pmatrix} 0 & \boldsymbol{e}_1^T \\ \boldsymbol{e}_0 & \boldsymbol{D}_0^{-1}\boldsymbol{C}_1 \end{pmatrix}\begin{pmatrix} 0 & \boldsymbol{e}_0^T \\ \boldsymbol{e}_0 & \boldsymbol{D}_0 \end{pmatrix}\begin{pmatrix} 1 & \boldsymbol{e}_0^T \\ -\boldsymbol{C}_1^{-1}\boldsymbol{e} & \boldsymbol{C}_1^{-1} \end{pmatrix} \\
&= \begin{pmatrix} -\boldsymbol{e}_1^T\boldsymbol{D}_0\boldsymbol{C}_1^{-1}\boldsymbol{e} & \boldsymbol{e}_1^T\boldsymbol{D}_0\boldsymbol{C}_1^{-1} \\ -\boldsymbol{D}_0^{-1}\boldsymbol{C}_1\boldsymbol{D}_0\boldsymbol{C}_1^{-1}\boldsymbol{e} & \boldsymbol{D}_0^{-1}\boldsymbol{C}_1\boldsymbol{D}_0\boldsymbol{C}_1^{-1} \end{pmatrix}. \qquad (3.12)
\end{aligned}$$

Combining Lemma 3.1 and (3.12) we have
(3.13)

$$(-\boldsymbol{D}_0^{-1}\boldsymbol{C}_1\boldsymbol{D}_0\boldsymbol{C}_1^{-1}\boldsymbol{e})_i = \frac{(-1)^i}{iC_i^k} \text{ and } (\boldsymbol{D}_0^{-1}\boldsymbol{C}_1\boldsymbol{D}_0\boldsymbol{C}_1^{-1})_{ij} = \begin{cases} \sum_{p=1}^{i}\frac{1}{p} - \sum_{p=1}^{k-i}\frac{1}{p}, & i=j, \\ \frac{(-1)^{i-j}}{i-j}\frac{C_j^k}{C_i^k}, & i\neq j. \end{cases}$$

Since $\boldsymbol{N} = \boldsymbol{D}_0(\boldsymbol{D}_0^{-1}\boldsymbol{C}_1\boldsymbol{D}_0\boldsymbol{C}_1^{-1})\boldsymbol{D}_0^{-1} + \boldsymbol{D}_0^{-1}$, it follows from (3.4), (3.8) and (3.13) that

$$(-\widetilde{\boldsymbol{b}})_i = \frac{(-1)^i}{C_i^k} \quad \text{and} \quad \boldsymbol{N}_{ij} = \begin{cases} \sum_{p=1}^{i}\frac{1}{p} - \sum_{p=1}^{k-i}\frac{1}{p} + \frac{1}{i}, & i=j, \\ \frac{(-1)^{i-j}}{i-j}\frac{i}{j}\frac{C_j^k}{C_i^k}, & i\neq j. \end{cases}$$

Let $\boldsymbol{v} = (1, 2^{k+1}, \cdots, k^{k+1})^T$, it follows from (2.5) and (3.4) that

$$\boldsymbol{C}_{p+1} = \frac{1}{(k+2)!}(\boldsymbol{A}\boldsymbol{D}_0\boldsymbol{v} - (p+1)\boldsymbol{B}\boldsymbol{v}).$$

By the definition of error in (2.12), letting $\boldsymbol{A} = \boldsymbol{I}$, we have $\boldsymbol{B} = \boldsymbol{N}^{-1}$ and

$$\boldsymbol{C}_{p+1} = \frac{1}{(k+2)!}(\boldsymbol{D}_0\boldsymbol{v} - (p+1)\boldsymbol{N}^{-1}\boldsymbol{v}).$$

This completes the proof. ∎

From Theorem 3.2, we observe that the stability function $R_{n+i}(z) = |\boldsymbol{C}_i(z)|/|\boldsymbol{C}(z)|$ is unique, so the stability of BIM is independent of the individual BIM matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. Next, we present the stability functions for the cases $k = 2, 3, \cdots, 8$ and discuss their stability properties in detail. For convenience of the reader, we list the coefficients of the stability functions in Table 10 in Appendix A. Without affecting the stability function, the coefficients of $|\boldsymbol{C}_i(z)|$ and $|\boldsymbol{C}(z)|$ are multiplied by a common factor such that both of them are integers.

THEOREM 3.3. *BIM has the following stability properties.*
(1) The methods are block k-step A-stable when $k = 2$ and $k = 3$.
(2) The methods are block k^{th}-step A-stable when $4 \leq k \leq 8$.

Proof. (1) It is clear that $k + 1 \geq 2k - 2$ when $k = 2$ and $k = 3$, and

$$\lim_{z\to\infty} R_{n+i}(z) = \lim_{z\to\infty}\frac{|\boldsymbol{C}_i(z)|}{|\boldsymbol{C}(z)|} \leq 1.$$

Moreover, the signs of the denominator $C(z)$ are alternating. From Theorem 2.2, we see that the methods are block k-step A-stable.

(2) If $k = 4$, since $\boldsymbol{C}(z) = \boldsymbol{C}_4(-z)$, we have

$$E_{n+4}(y) = |\boldsymbol{C}(\mathbf{i}y)|^2 - |\boldsymbol{C}_4(\mathbf{i}y)|^2 = \boldsymbol{C}(\mathbf{i}y)\boldsymbol{C}(-\mathbf{i}y) - \boldsymbol{C}_4(\mathbf{i}y)\boldsymbol{C}_4(-\mathbf{i}y) = 0 \;\; \forall y \in \mathbb{R}. \tag{3.14}$$

So $R_{n+4}(z)$ is I-stable. It is clear that $k+1 \geq 2k-3$ and the signs of the denominator $C(z)$ are alternating. From Theorem 2.3, we see that the methods are block k^{th}-step A-stable. Let $E_{n+i}(y) = |\boldsymbol{C}(\mathbf{i}y)|^2 - |\boldsymbol{C}_i(\mathbf{i}y)|^2$ $(i = 1, 2, 3)$, a straightforward

Table 2: The numerically computed roots of the function $|\boldsymbol{C}(z)|$

$k=7$	$k=8$
$0.2422+1.7552\mathbf{i}$	$0.1007+1.8078\mathbf{i}$
$0.2422-1.7552\mathbf{i}$	$0.1007-1.8078\mathbf{i}$
$0.7759+1.0735\mathbf{i}$	$0.6468+1.1811\mathbf{i}$
$0.7759-1.0735\mathbf{i}$	$0.6468-1.1811\mathbf{i}$
$1.1000+0.0000\mathbf{i}$	$0.9240+0.6841\mathbf{i}$
$1.0248+0.5199\mathbf{i}$	$0.9240-0.6841\mathbf{i}$
$1.0248-0.5199\mathbf{i}$	$1.0463+0.2249\mathbf{i}$
	$1.0463-0.2249\mathbf{i}$

computation shows that $E_{n+i}(y)\geq 0$ does not hold for some $y\in\mathbb{R}$. Therefore, $R_{n+i}(z)$ $(i=1,2,3)$ is not I-stable and the methods are not block k-step A-stable.

When $k=5,6,7,8$, it is easy to show that $R_{n+k}(z)$ is I-stable, but the above techniques are not valid. Let $z=\alpha+\mathbf{i}\beta$, if $k=5$, we have

$$\begin{aligned}&|\boldsymbol{C}(z)|^2-|\boldsymbol{C}_k(z)|^2\\&=-65760\alpha\beta^8+120(-2192\alpha^3-45\alpha)\beta^6+360(-1096\alpha^5-2765\alpha^3-630\alpha)\beta^4\\&+120(-2192\alpha^7-16455\alpha^5-22260\alpha^3-6300\alpha)\beta^2\\&-65760\alpha^9-984600\alpha^7-3826800\alpha^5-4644000\alpha^3-1296000\alpha.\end{aligned}$$

It is clear that

$$|\boldsymbol{C}(z)|^2\geq|\boldsymbol{C}_k(z)|^2$$

holds for all $\alpha\leq 0$. From Definition 2.1, we observe that $R_{n+k}(z)$ is A-stable.

If $k=6$, we obtain

$$\begin{aligned}&|\boldsymbol{C}(z)|^2-|\boldsymbol{C}_k(z)|^2\\&=-17640\alpha\beta^{10}+(-88200\alpha^3+25872\alpha)\beta^8+(-176400\alpha^5-249312\alpha^3-28140\alpha)\beta^6\\&+(-176400\alpha^7-903168\alpha^5-630420\alpha^3-120540\alpha)\beta^4\\&+(-88200\alpha^9-954912\alpha^7-2386020\alpha^5-1863960\alpha^3-352800\alpha)\beta^2\\&-17640\alpha^{11}-326928\alpha^9-1783740\alpha^7-3719100\alpha^5-2822400\alpha^3-529200\alpha.\end{aligned}\tag{3.15}$$

To show (3.15) is non-negative for all $\alpha\leq 0$, we first consider it's first three terms (factoring out β^6) denoted as $f(\beta)=-17640\alpha\beta^4+(-88200\alpha^3+25872\alpha)\beta^2-176400\alpha^5-249312\alpha^3-28140\alpha$, we compute the discriminant of the roots of the quadratic equation (in β^2),

$$\begin{aligned}\Delta&=(-88200\alpha^3+25872\alpha)^2-3\times 17640\alpha\times(176400\alpha^5+249312\alpha^3+28140\alpha)\\&=-4667544000\alpha^6-22155275520\alpha^4-1316198016\alpha^2\\&\leq 0,\end{aligned}\tag{3.16}$$

which implies $f(\beta)\geq 0$ for all $\alpha\leq 0$. Next consider the rest of the terms in (3.15) as a polynomial in β, we see that all coefficients are non-negative if $\alpha\leq 0$. Then, $|\boldsymbol{C}(z)|^2\geq|\boldsymbol{C}_k(z)|^2$ holds for all $\alpha\leq 0$, and this shows that the method is block k^{th}-step A-stable.

When $k=7$ and $k=8$, it is easy to see that $E_{n+k}(y)=|\boldsymbol{C}(\mathbf{i}y)|^2-|\boldsymbol{C}_k(\mathbf{i}y)|^2=0$ holds for all $y\in\mathbb{R}$ since $|\boldsymbol{C}(z)|=|\boldsymbol{C}_k(-z)|$, thus $R_{n+k}(z)$ is I-stable. Unfortunately, we are unable to prove that $R_{n+k}(z)$ is analytic by the above techniques. The roots of the function $|\boldsymbol{C}(z)|$ are computed numerically and shown in Table 2. It is clear that the real parts of all the roots are positive, therefore $R_{n+k}(z)$ is analytic and this method is block k^{th}-step A-stable. ∎

Theorem 3.3 shows that BIM is k^{th}-step A-stable when $4 \leq k \leq 8$, i.e., the functions $R_{n+i}(z)$ $(1 \leq i < k)$ are not A-stable. So it is interesting to study the stability of $R_{n+i}(z)$ $(1 \leq i < k)$, as an example, for the special case of $k = 8$, we present the stability regions in Figure 3. It is clear that the stability region of $R_{n+8}(z)$ coincides exactly with the entire negative half-plane $\mathbb{C}^-$, implying that this method is block k^{th}-step A-stable. Moreover, the subfigures (a) to (g) show that $R_{n+i}(z)$ $(1 \leq i \leq 7)$ is "nearly" A-stable. They are called $A(\alpha)$-stable in [53].

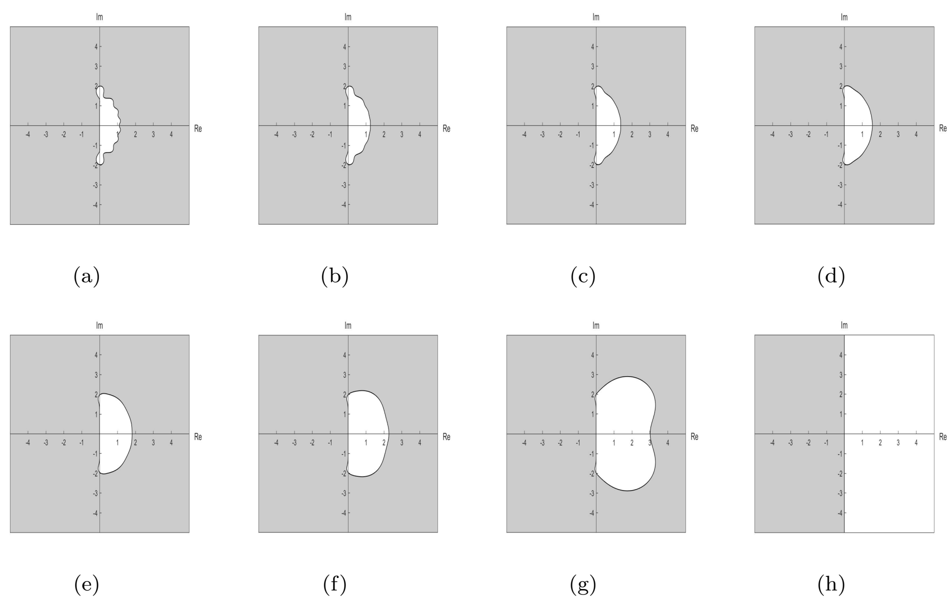

Fig. 3: The stability regions (gray) of BIM($k = 8$), $(a), (b), \cdots, (h)$ correspond to the stability function $R_{n+1}(z), R_{n+2}(z), \cdots, R_{n+8}(z)$, respectively

Remark 3.4. By employing the order conditions (2.5), we prove that the stability functions depend on $\boldsymbol{B}^{-1}\boldsymbol{A}$, and are independent of the individual matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. We also present explicit matrix form of $\boldsymbol{B}^{-1}\boldsymbol{A}$ which plays an important role in the stability analysis of BIM. From the order conditions (2.5), it is easy to see that the order of BIM with block size k is at least $k+1$. It has been proven that these methods converge at order $k + 1$ when k is odd and at order $k + 2$ when k is even [51, 52].

In Theorem 3.3, we give a thorough analysis for the stability of BIM with k up to 8. When $k = 9$, some roots of $|\boldsymbol{C}_k(z)|$ have negative real part, which means that this method is not block k^{th}-step A-stable. Because the roots are computed numerically for $k > 8$, we do not have a mathematically provable stability theory.

3.2. Construction of BIM with A-stability. In this section, we explicitly construct some BIM with A-stability. From (3.4), (3.7) and (3.9), we obtain

$$\left\{\begin{array}{ll} \boldsymbol{A} & = \boldsymbol{B}\boldsymbol{N}, \quad \boldsymbol{N} = \boldsymbol{C}_1\boldsymbol{D}_1\boldsymbol{C}_1^{-1}\boldsymbol{D}_0^{-1}, \\ \boldsymbol{a} & = -\boldsymbol{A}\boldsymbol{e} = -\boldsymbol{B}\boldsymbol{N}\boldsymbol{e}, \\ \boldsymbol{b} & = \boldsymbol{A}\widetilde{\boldsymbol{x}} - \boldsymbol{B}\boldsymbol{e} = \boldsymbol{B}\boldsymbol{N}\widetilde{\boldsymbol{x}} - \boldsymbol{B}\boldsymbol{e}, \end{array}\right. \tag{3.17}$$

where $\widetilde{\boldsymbol{x}} = (1, 2, \cdots, k)^T$, $\boldsymbol{N}$ is given explicitly in (3.1). Hence, if $\boldsymbol{A}$ (or $\boldsymbol{B}$) is given, then $\boldsymbol{B}$ (or $\boldsymbol{A}$), $\boldsymbol{a}$ and $\boldsymbol{b}$ can also be fixed. The question is how to choose $\boldsymbol{A}$ and $\boldsymbol{B}$ such that the corresponding BIM has desirable properties? In the following, we focus on how to choose $\boldsymbol{A}$ and $\boldsymbol{B}$ in terms of the positive definiteness and sparsity.

DEFINITION 3.5. *A matrix $\boldsymbol{N}$ is said to be positive stable if all its eigenvalues have positive real parts.*

LEMMA 3.6. *([39]) $\boldsymbol{N}$ is positive stable if and only if there exists a symmetric positive definite (SPD) matrix $\boldsymbol{B}$ such that*

$$\boldsymbol{B}\boldsymbol{N} + \boldsymbol{N}^T\boldsymbol{B} \tag{3.18}$$

is positive definite.

Especially, if (3.18) is positive definite with a positive diagonal matrix $\boldsymbol{B}$, $\boldsymbol{N}$ is also called diagonally stable or "Lyapunov diagonally stable" [3, 25].

LEMMA 3.7. *([3]) A matrix $\boldsymbol{N}$ is diagonally stable if and only if for every nonzero positive semidefinite matrix $\boldsymbol{B}$, $\boldsymbol{B}\boldsymbol{N}$ has a positive diagonal element.*

Lemma 3.7 provides a necessary condition for a diagonally stable $\boldsymbol{N}$ such that all the diagonal elements of $\boldsymbol{N}$ are positive. From Theorem 3.2, the matrix $\boldsymbol{N}$ for different k $(2 \leq k \leq 8)$ can be given explicitly as

$$\boldsymbol{N}_2 = \begin{pmatrix} 1 & 1/4 \\ -4 & 2 \end{pmatrix}, \ \boldsymbol{N}_3 = \begin{pmatrix} 1/2 & 1/2 & -1/18 \\ -2 & 1 & 2/9 \\ 9/2 & -9/2 & 13/6 \end{pmatrix}, \ \boldsymbol{N}_4 = \begin{pmatrix} 1/6 & 3/4 & -1/6 & 1/48 \\ -4/3 & 1/2 & 4/9 & -1/24 \\ 3/2 & -9/4 & 7/6 & 3/16 \\ -16/3 & 6 & -16/3 & 7/3 \end{pmatrix},$$

$$\boldsymbol{N}_5 = \begin{pmatrix} -1/12 & 1 & -1/3 & 1/12 & -1/100 \\ -1 & 1/6 & 2/3 & -1/8 & 1/75 \\ 3/4 & -3/2 & 2/3 & 3/8 & -3/100 \\ -4/3 & 2 & -8/3 & 4/3 & 4/25 \\ 25/4 & -25/3 & 25/3 & -25/4 & 149/60 \end{pmatrix},$$

$$\boldsymbol{N}_6 = \begin{pmatrix} -17/60 & 5/4 & -5/9 & 5/24 & -1/20 & 1/180 \\ -4/5 & -1/12 & 8/9 & -1/4 & 4/75 & -1/180 \\ 9/20 & -9/8 & 1/3 & 9/16 & -9/100 & 1/120 \\ -8/15 & 1 & -16/9 & 5/6 & 8/25 & -1/45 \\ 5/4 & -25/12 & 25/9 & -25/8 & 89/60 & 5/36 \\ -36/5 & 45/4 & -40/3 & 45/4 & -36/5 & 157/60 \end{pmatrix},$$

$$\boldsymbol{N}_7 = \begin{pmatrix} -9/20 & 3/2 & -5/6 & 5/12 & -3/20 & 1/30 & -1/294 \\ -2/3 & -17/60 & 10/9 & -5/12 & 2/15 & -1/36 & 2/735 \\ 3/10 & -9/10 & 1/12 & 3/4 & -9/50 & 1/30 & -3/980 \\ -4/15 & 3/5 & -4/3 & 1/2 & 12/25 & -1/15 & 4/735 \\ 5/12 & -5/6 & 25/18 & -25/12 & 59/60 & 5/18 & -5/294 \\ -6/5 & 9/4 & -10/3 & 15/4 & -18/5 & 97/60 & 6/49 \\ 49/6 & -147/10 & 245/12 & -245/12 & 147/10 & -49/6 & 383/140 \end{pmatrix},$$

$$
\boldsymbol{N}_8 = \begin{pmatrix}
-83/140 & 7/4 & -7/6 & 35/48 & -7/20 & 7/60 & -1/42 & 1/448 \\
-4/7 & -9/20 & 4/3 & -5/8 & 4/15 & -1/12 & 4/245 & -1/672 \\
3/14 & -3/4 & -7/60 & 15/16 & -3/10 & 1/12 & -3/196 & 3/2240 \\
-16/105 & 2/5 & -16/15 & 1/4 & 16/25 & -2/15 & 16/735 & -1/560 \\
5/28 & -5/12 & 5/6 & -25/16 & 13/20 & 5/12 & -5/98 & 5/1344 \\
-12/35 & 3/4 & -4/3 & 15/8 & -12/5 & 67/60 & 12/49 & -3/224 \\
7/6 & -49/20 & 49/12 & -245/48 & 49/10 & -49/12 & 243/140 & 7/64 \\
-64/7 & 56/3 & -448/15 & 35 & -448/15 & 56/3 & -64/7 & 199/70
\end{pmatrix},
$$

respectively. For the above given $\boldsymbol{N}$, we present three types of $\boldsymbol{B}$ and $\boldsymbol{A}$:

(1) $\boldsymbol{B}$ is chosen as a SPD matrix such that $\boldsymbol{A} = \boldsymbol{BN}$ is positive definite.

From Lemma 3.6, we see that there exists a SPD matrix $\boldsymbol{B}$ such that $\boldsymbol{BN}$ is positive definite, but the matrix $\boldsymbol{B}$ is difficult to find for a given $\boldsymbol{N}$.

(2) $\boldsymbol{B}$ is chosen as a positive diagonal matrix such that $\boldsymbol{A} = \boldsymbol{BN}$ is positive definite.

Generally speaking, there may not exist a positive diagonal matrix $\boldsymbol{B}$ for a given $\boldsymbol{N}$ such that $\boldsymbol{BN}$ is positive definite. Lemma 3.7 provides a sufficient and necessary condition, there is no obvious way to explicitly find such a positive diagonal matrix. The theoretical characterization of the class of diagonally stable matrices is not computationally effective. Some optimization-based numerical algorithms were developed in [4, 20, 27, 33]. Below, we present some computed $\boldsymbol{B}$ by using the interior point methods in [4]. For $k = 2, 3, 4$, the corresponding $\boldsymbol{B}$ are given as

$$
\boldsymbol{B}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1/2 \end{pmatrix}, \quad \boldsymbol{B}_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1/10 \end{pmatrix}, \quad \boldsymbol{B}_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 3/4 & 0 & 0 \\ 0 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 1/10 \end{pmatrix},
$$

respectively. Note that when such $\boldsymbol{B}$ exists, it is not unique. From Lemma 3.7, we see that there exists no $\boldsymbol{B}$ for the given $\boldsymbol{N}$ when $5 \le k \le 8$.

(3) $\boldsymbol{B}$ is chosen as an identity matrix such that $\boldsymbol{A} = \boldsymbol{N}$.

It is trivial to choose such matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ since $\boldsymbol{N}$ can be given explicitly, however the matrix $\boldsymbol{A}$ is only positive stable but not positive definite.

Below we present some BIM with A-stability.

Algorithm 3.1 (BIM with A-stability)

Let $\boldsymbol{N}$ be defined by $\boldsymbol{N}_k$ for $2 \le k \le 8$, all methods defined by the following BIM matrices and vectors

$$
\begin{cases} \boldsymbol{B} = \boldsymbol{B}_k, \ \boldsymbol{A} = \boldsymbol{BN}_k, \ \boldsymbol{a} = -\boldsymbol{Ae}, \ \boldsymbol{b} = \boldsymbol{A}\widetilde{\boldsymbol{x}} - \boldsymbol{Be}, & \text{for} \quad k = 2, 3, 4, \\ \boldsymbol{B} = \boldsymbol{I}, \quad \boldsymbol{A} = \boldsymbol{BN}_k, \ \boldsymbol{a} = -\boldsymbol{Ae}, \ \boldsymbol{b} = \boldsymbol{A}\widetilde{\boldsymbol{x}} - \boldsymbol{Be}, & \text{for} \quad k = 5, 6, 7, 8, \end{cases} \tag{3.19}
$$

are A-stable.

---

4. Comparisons with some fully implicit Runge-Kutta methods. The s-stage IRK is an important class of time-integration schemes and has been studied extensively since they can offer high order accuracy and excellent stability [9,50]. IRK can be characterized by the Butcher tableau

$$
\begin{array}{c|c} \widehat{\boldsymbol{c}} & \widehat{\boldsymbol{A}} \\ \hline & \widehat{\boldsymbol{b}}^T \end{array},
$$

where the Runge-Kutta matrix $\widehat{\boldsymbol{A}}$ is positive stable and dense, but not positive definite. When applied to PDEs, one needs to solve a large, strongly coupled linear

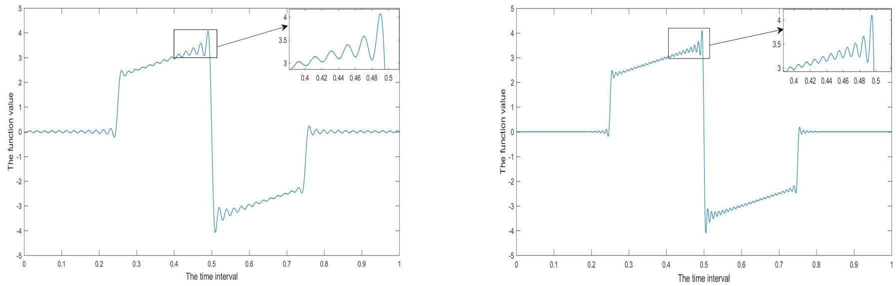

Fig. 4: Exact solutions for $M = 50$ (left) and $M = 100$ (right)

Table 3: The errors of BIM($k = 2, p = 4$), BIM($k = 3, p = 4$) and FIRK (Gauss(2)) for solving Example 3 with $M = 50$, N_f is the number of function evaluations

	BIM($k=2,p=4$)			BIM($k=3,p=4$)			FIRK		
N_f	$1/\tau$	$\|\boldsymbol{y}_e - \boldsymbol{y}\|$	order	$1/\tau$	$\|\boldsymbol{y}_e - \boldsymbol{y}\|$	order	$1/\tau$	$\|\boldsymbol{y}_e - \boldsymbol{y}\|$	order
48	48	1.15e+0		48	1.05e+0		24	3.69e+0	
96	96	6.86e-1	0.74	96	9.98e-1	0.07	48	1.09e+0	1.75
192	192	8.49e-2	3.01	192	5.63e-2	4.15	96	9.12e-2	3.57
384	384	4.68e-3	4.18	384	3.02e-3	4.22	192	7.33e-3	3.64
768	768	3.11e-4	3.91	768	1.80e-4	4.07	384	4.64e-4	3.98
1536	1536	1.99e-5	3.97	1536	1.06e-5	4.09	768	3.00e-5	3.95
3072	3072	1.24e-6	4.00	3072	6.15e-7	4.10	1536	1.87e-6	4.00
6144	6144	7.76e-8	4.00	6144	3.71e-8	4.05	3072	1.17e-7	4.00

system at every time step. Because the lack of positive definiteness of $\widehat{\boldsymbol{A}}$, the linear system is often difficult to precondition and solve by iterative methods. On the other hand, for BIM the stability functions are independent of the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, so it is possible to construct different methods with the desired order and stability properties. Algorithms 3.1 provides schemes with up to 8^{th} order of accuracy. For these schemes, $\boldsymbol{B}$ can be chosen as a positive diagonal matrix or an identity matrix, and $\boldsymbol{A}$ is positive definite for A-stable algorithm with $2 \leq k \leq 4$.

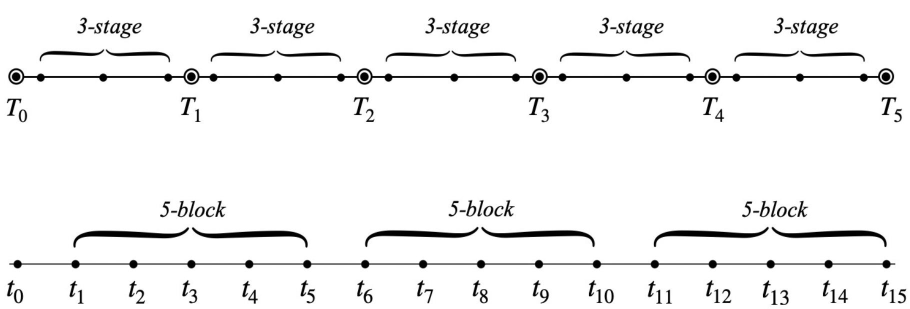

Fig. 5: FIRK (Gauss(3) (top) and BIM($k = 5, p = 6$) (bottom), ($T_0 = t_0$, $T_1 = t_3$, $T_2 = t_6$, $T_3 = t_9$, $T_4 = t_{12}$, $T_5 = t_{15}$)

Table 4: The errors of BIM($k=4,p=6$), BIM($k=5,p=6$) and FIRK(Gauss(3)) for solving Example 3 with $M=50$, N_f is the number of function evaluations

	BIM($k=4,p=6$)			BIM($k=5,p=6$)			FIRK		
N_f	$1/\tau$	$\lvert\boldsymbol{y}_e-\boldsymbol{y}\rvert$	order	$1/\tau$	$\lvert\boldsymbol{y}_e-\boldsymbol{y}\rvert$	order	$1/\tau$	$\lvert\boldsymbol{y}_e-\boldsymbol{y}\rvert$	order
60	60	1.31e+0		60	5.64e-1		20	1.52e+0	
120	120	5.41e-1	1.28	120	7.78e-1	-0.46	40	3.06e-1	2.31
240	240	2.20e-2	4.62	240	2.01e-2	5.28	80	1.29e-2	4.57
480	480	3.98e-4	5.79	480	7.82e-5	8.00	160	2.36e-4	5.77
960	960	4.53e-6	6.46	960	2.22e-6	5.14	320	3.78e-6	5.97
1920	1920	1.01e-7	5.48	1920	3.11e-8	6.16	640	5.98e-8	5.98
3840	3840	1.51e-9	6.06	3840	4.60e-10	6.08	1280	9.48e-10	5.98
7680	7680	2.39e-11	5.98	7680	6.77e-12	6.08	2560	1.48e-11	6.00

Table 5: The errors of BIM($k=2,p=4$), BIM($k=3,p=4$) and FIRK(Gauss(2)) for solving Example 3 with $M=100$, N_f is the number of function evaluations

	BIM($k=2,p=4$)			BIM($k=3,p=4$)			FIRK		
N_f	$1/\tau$	$\lvert\boldsymbol{y}_e-\boldsymbol{y}\rvert$	order	$1/\tau$	$\lvert\boldsymbol{y}_e-\boldsymbol{y}\rvert$	order	$1/\tau$	$\lvert\boldsymbol{y}_e-\boldsymbol{y}\rvert$	order
48	48	3.07e+0		48	2.91e+0		24	4.19e+0	
96	96	1.64e+0	0.90	96	1.45e+0	1.00	48	3.22e+0	0.38
192	192	8.77e-1	0.90	192	1.13e+0	0.36	96	4.75e-1	2.75
384	384	5.81e-2	3.92	384	5.56e-2	4.35	192	4.75e-2	3.32
768	768	3.04e-3	4.25	768	2.98e-3	4.22	384	2.80e-3	4.08
1536	1536	2.08e-4	3.87	1536	1.80e-4	4.05	768	1.71e-4	4.03
3072	3072	1.29e-5	4.01	3072	1.12e-5	4.00	1536	1.06e-5	4.00
6144	6144	8.14e-7	3.99	6144	6.99e-7	4.00	3072	6.62e-7	4.00

In this section, we compare the s-stage FIRK algorithms (Gauss-Legendre and BIM for solving the following ODE.

EXAMPLE 3. $y' = -250y + f(t), \quad t \in (0,1], \quad y(0) = 0.$

Here $f(t)$ is chosen such that the exact solution is

$$y(t) = \sum_{m=1}^{M} b_m \sin 2m\pi t,$$

where $b_m = \frac{9}{2m\pi}(\cos\frac{m\pi}{2}\cos m\pi) + \frac{5}{2m^2\pi^2}(\sin\frac{m\pi}{2}\cos m\pi) - \frac{7}{m\pi}\cos m\pi$. Figure 4 shows that the smoothness of the solution becomes worse when M increases. In the experiments, we will use two different values of $M = 50, 100$.

Since the main computational costs for these two methods are the evaluations of the functions and solving the linear systems, we compare the errors and the order of convergence by setting the same number, denoted as N_f, of function evaluations or the number of linear systems required to solve. In Tables 3-6, $|\boldsymbol{y}_e - \boldsymbol{y}|$ denotes the maximum norm of the error between the exact solution $\boldsymbol{y}_e$ and the numerical solution $\boldsymbol{y}$ in the entire time interval. We see that none of the algorithms is able to achieve the optimal order if the time step size τ is too large, the reason is that the solution is not smooth in part of the interval as shown in Figure 4. When the time step size τ decreases, it is clear that all the algorithms converge at the optimal order. Table 3 show that the errors of BIM($k=2,p=4$) and BIM($k=3,p=4$) are smaller than Gauss(2). If we choose larger $M = 100$, the numerical results in Table 5 show

Table 6: The errors of BIM($k=4, p=6$), BIM($k=5, p=6$) and FIRK(Gauss(3)) for solving Example 3 with $M=100$, N_f is the number of function evaluations

	BIM($k=4,p=6$)			BIM($k=5,p=6$)			FIRK		
N_f	$1/\tau$	$\lvert \boldsymbol{y}_e-\boldsymbol{y}\rvert$	order	$1/\tau$	$\lvert \boldsymbol{y}_e-\boldsymbol{y}\rvert$	order	$1/\tau$	$\lvert \boldsymbol{y}_e-\boldsymbol{y}\rvert$	order
60	60	3.45e+0		60	1.53e+0		20	4.87e+0	
120	120	1.13e+0	1.60	120	9.77e-1	0.64	40	5.63e-1	3.11
240	240	5.62e-1	1.01	240	7.44e-1	0.39	80	1.70e-1	1.72
480	480	2.94e-2	4.26	480	2.13e-2	5.13	160	1.03e-3	7.37
960	960	4.98e-4	5.88	960	8.02e-5	8.05	320	4.17e-5	4.63
1920	1920	7.18e-6	6.12	1920	2.02e-6	5.31	640	6.12e-7	6.09
3840	3840	1.11e-7	6.02	3840	3.26e-8	5.96	1280	9.72e-9	5.98
7680	7680	1.72e-9	6.01	7680	4.82e-10	6.08	2560	1.52e-10	6.00

that BIM are almost the same as that of FIRK in terms of the errors. From the numerical results in Tables 4 and 6 we observe that the error of BIM($k=5, p=6$) is a little smaller than that of FIRK (Gauss(3)) algorithms when $M=50$, and the errors are almost the same in the case $M=100$. Tables 4 and 6 also show that Gauss(3) performs a little better than BIM($k=4, p=6$) in the case both $M=50$ and $M=100$.

In Figure 5, we compare a different aspect of FIRK (Gauss(3)) and BIM($k=5, p=6$). We observe that in a single time step Gauss(3) produces one solution at the cost of three function evaluations at three not equally spaced temporal locations, BIM($k=5, p=6$) produces five solutions at five equally spaced temporal locations. It is clear that FIRK and BIM have the same desirable properties of high order accuracy, good stability and requiring one starting value. However, k in BIM is usually larger than s in FIRK to achieve the same order of convergence. This means that a larger system needs to be solved for BIM at a time, although the number of blocks of BIM is fewer than FIRK at the final time; see Figure 5. Fortunately, the coefficient matrix of BIM is positive definite if the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are properly chosen, and as a result, the systems are easier to solve. Moreover, BIM produces solutions at more temporal locations, that might be useful for certain applications.

5. BIM for parabolic PDEs. In this section, we study the proposed BIM for parabolic PDEs. We provide a convergence theory and some *a priori* error estimates for a model problem whose discretization consisting of BIM in time and the regular finite element in space. In order to solve the large systems resulting from the discretization, some parallel DD preconditioners are also introduced and studied.

5.1. A model problem and its finite element discretization. We consider a model parabolic equation

$$\left\{\begin{array}{rll} u_t - \nabla\cdot(a(x)\nabla u) & = f, & \text{in } \Omega\times(0,T], \\ u(x,t) & = 0, & \text{on } \partial\Omega\times(0,T], \\ u(x,0) & = u_0(x), & \text{in } \Omega, \end{array}\right. \tag{5.1}$$

where $\Omega\subset R^d$ ($d=2$ or 3) is a bounded, open polygonal (or polyhedra) domain, $0<a_0\le a(x)<+\infty$ and $f(x,t)\in L^2(\Omega\times(0,T])$, $u_0(x)\in L^2(\Omega)$. For simplicity, we set $a(x)\equiv 1$ in some of the following discussions.

Let $0=t_0<t_1<\cdots<t_m=T$ be a uniform temporal mesh and $\tau=t_i-t_{i-1}$. Suppose $u^i=u(x,t_i)$ is the solution at time t_i. Traditional time-stepping methods solve (5.1) time step by time step. We consider the method outlined in (2.12) for the

time discretization of (5.1) with block size $k \le m$ at time $t_1, t_2, \cdots, t_k$, that is to find $u^i \in H_0^1(\Omega)$, $i = 1, 2, \cdots, k$, such that

$$
\left\{
\begin{aligned}
\sum_{j=1}^{k} a_{1j}(u^j, v^1) + \tau \sum_{j=1}^{k} b_{1j}(\nabla u^j, \nabla v^1) &= (g^1, v^1),\\
\sum_{j=1}^{k} a_{2j}(u^j, v^2) + \tau \sum_{j=1}^{k} b_{2j}(\nabla u^j, \nabla v^2) &= (g^2, v^2),\\
&\vdots\\
\sum_{j=1}^{k} a_{kj}(u^j, v^k) + \tau \sum_{j=1}^{k} b_{kj}(\nabla u^j, \nabla v^k) &= (g^k, v^k),
\end{aligned}
\right.
\tag{5.2}
$$

where

$$
(g^i, v^i) = -a_{i0}(u^0, v^i) + \tau b_{i0}(f^0, v^i) - \tau b_{i0}(\nabla u^0, \nabla v^i) + \tau \sum_{j=1}^{k} b_{ij}(f^j, v^i) \quad \forall v^i \in H_0^1(\Omega).
$$

Let $\boldsymbol{u} = (u^1, u^2, \cdots, u^k)^T \in (H_0^1(\Omega))^k$, $\boldsymbol{f} = (f^1, f^2, \cdots, f^k)^T \in (L^2(\Omega \times [0, T]))^k$. For any $\boldsymbol{v} \in (H_0^1(\Omega))^k$, the equivalent variational form of (5.2) is

$$
a_\tau(\boldsymbol{u}, \boldsymbol{v}) \equiv (\boldsymbol{A}\boldsymbol{u}, \boldsymbol{v}) + \tau(\boldsymbol{B}\nabla\boldsymbol{u}, \nabla\boldsymbol{v}) = (\boldsymbol{g}, \boldsymbol{v}),
\tag{5.3}
$$

where $\nabla\boldsymbol{u} = (\nabla u^1, \nabla u^2, \cdots, \nabla u^k)^T$ and

$$
(\boldsymbol{g}, \boldsymbol{v}) = \tau(\boldsymbol{B}\boldsymbol{f}, \boldsymbol{v}) - (\boldsymbol{a} \otimes u^0, \boldsymbol{v}) + \tau(\boldsymbol{b} \otimes f^0, \boldsymbol{v}) - \tau(\boldsymbol{b} \otimes \nabla u^0, \nabla\boldsymbol{v}).
$$

Next, we present the boundedness of the bilinear form $a_\tau(\cdot, \cdot)$ defined in (5.3) from above and below under the $\|\cdot\|_\tau$ norm for different matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. The estimates are summarized in the following four lemmas.

LEMMA 5.1. *Suppose that $\boldsymbol{B}$ is SPD and $\boldsymbol{A}$ is positive definite, there exist positive constants C_0 and c_0 independent of τ such that*

$$
a_\tau(\boldsymbol{u}, \boldsymbol{v}) \le C_0 \|\boldsymbol{u}\|_\tau \|\boldsymbol{v}\|_\tau \quad \forall \boldsymbol{u}, \boldsymbol{v} \in (H_0^1(\Omega))^k
$$

and

$$
a_\tau(\boldsymbol{u}, \boldsymbol{u}) \ge c_0 \|\boldsymbol{u}\|_\tau^2 \quad \forall \boldsymbol{u} \in (H_0^1(\Omega))^k,
$$

where $\|\boldsymbol{v}\|_\tau^2 = \|\boldsymbol{v}\|^2 + \tau\|\nabla\boldsymbol{v}\|^2$.

Proof. It follows from the Cauchy-Schwarz inequality that

$$
\begin{aligned}
a_\tau(\boldsymbol{u},\boldsymbol{v}) &= (\boldsymbol{A}\boldsymbol{u},\boldsymbol{v}) + \tau(\boldsymbol{B}\nabla\boldsymbol{u},\nabla\boldsymbol{v}) \\
&= \sum_{i,j=1}^{k} a_{ij}(u^i,v^j) + \tau\sum_{i,j=1}^{k} b_{ij}(\nabla u^i,\nabla v^j) \\
&\le a_{max}\sum_{i,j=1}^{k}\|u^i\|\|v^j\| + \tau b_{max}\sum_{i,j=1}^{k}\|\nabla u^i\|\|\nabla v^j\| \\
&\le C_0(\|\boldsymbol{u}\|\|\boldsymbol{v}\| + \tau\|\nabla\boldsymbol{u}\|\|\nabla\boldsymbol{v}\|) \\
&\le C_0(\|\boldsymbol{u} + \tau\|\nabla\boldsymbol{u}\|)^{\frac{1}{2}}(\|\boldsymbol{v} + \tau\|\nabla\boldsymbol{v}\|)^{\frac{1}{2}} \\
&= C_0\|\boldsymbol{u}\|_\tau\|\boldsymbol{v}\|_\tau, \qquad (5.4)
\end{aligned}
$$

where $a_{max} = \max_{i,j=1}^{k}|a_{ij}|$, $b_{max} = \max_{i,j=1}^{k}|b_{ij}|$ and $C_0 = k\max\{a_{max}, b_{max}\}$.

Further, since $\boldsymbol{B}$ is SPD and $\boldsymbol{A}$ is positive definite, we have

$$
\begin{aligned}
a_\tau(\boldsymbol{u},\boldsymbol{u}) &= (\boldsymbol{A}\boldsymbol{u},\boldsymbol{u}) + \tau(\boldsymbol{B}\nabla\boldsymbol{u},\nabla\boldsymbol{u}) \\
&\ge \frac{1}{2}\lambda_{min}(\boldsymbol{A}+\boldsymbol{A}^T)\|\boldsymbol{u}\|^2 + \tau\lambda_{min}(\boldsymbol{B})\|\nabla\boldsymbol{u}\|^2 \\
&\ge c_0\|\boldsymbol{u}\|_\tau^2, \qquad (5.5)
\end{aligned}
$$

where $c_0 = \min\{\frac{1}{2}\lambda_{min}(\boldsymbol{A}+\boldsymbol{A}^T), \lambda_{min}(\boldsymbol{B})\}$. ◻

Lemma 5.2. *Suppose that $\boldsymbol{B}$ is SPD and $\boldsymbol{A}$ is not positive definite, we have*

$$
a_\tau(\boldsymbol{u},\boldsymbol{v}) \le C_0\|\boldsymbol{u}\|_\tau\|\boldsymbol{v}\|_\tau \quad \forall \boldsymbol{u},\boldsymbol{v} \in (H_0^1(\Omega))^k
$$

and

$$
a_\tau(\boldsymbol{u},\boldsymbol{u}) \ge c_1\tau\|\boldsymbol{u}\|_1^2 - c_2\|\boldsymbol{u}\|^2 \quad \forall \boldsymbol{u} \in (H_0^1(\Omega))^k,
$$

where C_0, c_1 and c_2 are positive constants and independent of τ.

Proof. The upper bound has been estimated in (5.4). Since $\boldsymbol{B}$ is SPD and $\boldsymbol{A}$ is not positive definite, the smallest eigenvalue of $\frac{1}{2}(\boldsymbol{A}+\boldsymbol{A}^T)$ is negative, we have

$$
\begin{aligned}
a_\tau(\boldsymbol{u},\boldsymbol{u}) &= (\boldsymbol{A}\boldsymbol{u},\boldsymbol{u}) + \tau(\boldsymbol{B}\nabla\boldsymbol{u},\nabla\boldsymbol{u}) \\
&\ge \frac{1}{2}\lambda_{min}(\boldsymbol{A}+\boldsymbol{A}^T)\|\boldsymbol{u}\|^2 + \tau\lambda_{min}(\boldsymbol{B})\|\nabla\boldsymbol{u}\|^2 \\
&= c_1\|\boldsymbol{u}\|_\tau^2 - (c_2+c_1)\|\boldsymbol{u}\|^2, \qquad (5.6)
\end{aligned}
$$

where $c_1 = \lambda_{min}(\boldsymbol{B})$ and $c_2 = -\frac{1}{2}\lambda_{min}(\boldsymbol{A}+\boldsymbol{A}^T)$. ◻

Lemma 5.3. *Suppose that $\boldsymbol{B}$ is positive stable and $\boldsymbol{A}$ is an identity matrix, there exist positive constants C_0 and c_3 independent of τ such that*

$$
a_\tau(\boldsymbol{u},\boldsymbol{v}) \le C_0\|\boldsymbol{u}\|_\tau\|\boldsymbol{v}\|_\tau \quad \forall \boldsymbol{u},\boldsymbol{v} \in \boldsymbol{V}
$$

and

$$
\sup_{\boldsymbol{v}\in\boldsymbol{V}} \frac{a_\tau(\boldsymbol{u},\boldsymbol{v})}{\|\boldsymbol{v}\|_\tau} \ge c_3\|\boldsymbol{u}\|_\tau \quad \forall \boldsymbol{u} \in \boldsymbol{V},
$$

where $\boldsymbol{V} = (L^2(\Omega))^k + \tau(H_0^1(\Omega))^k$. Moreover, there exists $\boldsymbol{u} \in \boldsymbol{V}$ such that $a_\tau(\boldsymbol{u},\boldsymbol{v}) \neq 0$ for any $\boldsymbol{v} \in \boldsymbol{V}$.

Proof. See Lemma 3.1 and Lemma 3.3 in [40]. ∎

Let $\boldsymbol{V}_h \in (H_0^1(\Omega))^k$ be the piecewise linear continuous finite element space. The finite element solution of (5.3) is to find $\boldsymbol{u}_h \in \boldsymbol{V}_h$ such that

$$a_\tau(\boldsymbol{u}_h, \boldsymbol{v}_h) = (\boldsymbol{g}, \boldsymbol{v}_h) \quad \forall \boldsymbol{v}_h \in \boldsymbol{V}_h. \tag{5.7}$$

THEOREM 5.4. *Suppose that $\boldsymbol{B}$ is SPD, there exists a positive constant C such that*

$$\|\boldsymbol{u} - \boldsymbol{u}_h\| \le \frac{Ch}{\sqrt{h^2+\tau}} \|\boldsymbol{u} - \boldsymbol{u}_h\|_\tau \quad \boldsymbol{u}_h \in \boldsymbol{V}_h.$$

Further, (1) if $\boldsymbol{A}$ is positive definite, (5.7) has a unique solution and

$$\|\boldsymbol{u} - \boldsymbol{u}_h\|_\tau \le C\|\boldsymbol{u}\|_\tau,$$

(2) if $\boldsymbol{A}$ is not positive definite, but h^2/τ is sufficiently small, (5.7) has a unique solution and

$$\|\boldsymbol{u} - \boldsymbol{u}_h\|_\tau \le C\|\boldsymbol{u}\|_\tau,$$

where $C > 0$ is independent of h and τ.

Proof. From the a priori error estimate of the finite element method and the definition of the τ-norm, we have

$$\|\boldsymbol{u} - \boldsymbol{u}_h\|^2 \le Ch^2|\boldsymbol{u} - \boldsymbol{u}_h|_1^2 \le \frac{Ch^2}{\tau}\|\boldsymbol{u} - \boldsymbol{u}_h\|_\tau^2 - \frac{Ch^2}{\tau}\|\boldsymbol{u} - \boldsymbol{u}_h\|^2,$$

which implies

$$\|\boldsymbol{u} - \boldsymbol{u}_h\| \le \frac{Ch}{\sqrt{h^2+\tau}}\|\boldsymbol{u} - \boldsymbol{u}_h\|_\tau. \tag{5.8}$$

If $\boldsymbol{A}$ is positive definite, it follows from (5.4) and (5.5) that

$$a_\tau(\boldsymbol{u} - \boldsymbol{u}_h, \boldsymbol{u}) \le C\|\boldsymbol{u} - \boldsymbol{u}_h\|_\tau \|\boldsymbol{u}\|_\tau,$$

and

$$a_\tau(\boldsymbol{u} - \boldsymbol{u}_h, \boldsymbol{u}) = a_\tau(\boldsymbol{u} - \boldsymbol{u}_h, \boldsymbol{u} - \boldsymbol{u}_h) \ge c_0\|\boldsymbol{u} - \boldsymbol{u}_h\|_\tau^2.$$

It is obvious that $\|\boldsymbol{u} - \boldsymbol{u}\|_\tau \le C\|\boldsymbol{u}_h\|_\tau$.

If $\boldsymbol{A}$ is not positive definite, it follows from (5.6) and (5.8) that

$$\begin{aligned} a_\tau(\boldsymbol{u} - \boldsymbol{u}_h, \boldsymbol{u}) = a_\tau(\boldsymbol{u} - \boldsymbol{u}_h, \boldsymbol{u} - \boldsymbol{u}_h) &\ge c_1\|\boldsymbol{u} - \boldsymbol{u}_h\|_\tau^2 - (c_1 + c_2)\|\boldsymbol{u} - \boldsymbol{u}_h\|^2 \\ &\ge \left(c_1 - \frac{Ch^2}{\tau + h^2}\right)\|\boldsymbol{u} - \boldsymbol{u}_h\|_\tau^2. \end{aligned}$$

Suppose h^2/τ is sufficiently small, we have $\|\boldsymbol{u} - \boldsymbol{u}_h\|_\tau \le C\|\boldsymbol{u}\|_\tau$, where $C > 0$ is independent of h and τ. ∎

Discretized by the finite element basis functions ϕ_j $(1 \le j \le N)$, the bilinear form (5.7) is equivalent to the following linear system of algebraic equations

$$\boldsymbol{\mathcal{A}}\,\boldsymbol{\mathcal{U}} \equiv (\boldsymbol{A} \otimes \boldsymbol{M} + \tau \boldsymbol{B} \otimes \boldsymbol{K})\,\boldsymbol{\mathcal{U}} = \boldsymbol{\mathcal{F}}, \quad \boldsymbol{\mathcal{A}} \in \mathbb{R}^{kN\times kN},\ \boldsymbol{\mathcal{F}} \in \mathbb{R}^{kN}, \tag{5.9}$$

where $\boldsymbol{A} \in \mathbb{R}^{k\times k}$ and $\boldsymbol{B} \in \mathbb{R}^{k\times k}$ are the BIM matrices, $\boldsymbol{M} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{K} \in \mathbb{R}^{N\times N}$ denote the mass matrix and the stiffness matrix, respectively. $\boldsymbol{\mathcal{U}}$ and $\boldsymbol{\mathcal{F}}$ are vectors corresponding to the nodal values of $\boldsymbol{u}_h$ and $\boldsymbol{g}$.

When BIM is used for PDEs, it results in a $kN \times kN$ linear system (5.9). In practice, N is usually large and $\boldsymbol{K}$ is highly ill-conditioned, a preconditioner is important if the system is solved by a Krylov subspace method.

5.2. Parallel preconditioning techniques. Note that (5.9) also arises from the classical implicit Runge-Kutta methods, in such a situation $\boldsymbol{A}$ is an identity matrix and $\boldsymbol{B}$ is the Runge-Kutta matrix which is dense, non-symmetric and not positive definite. Depending on how $\boldsymbol{B}$ is approximated, several preconditioners are available [28–30, 35, 40, 43, 47]. Different from the Runge-Kutta algorithms, for BIM, the matrix $\boldsymbol{B}$ in (5.9) is a positive diagonal matrix or an identity matrix, the matrix $\boldsymbol{A}$ is dense and positive definite or positive stable. Now we present some preconditioners for BIM based on the partition of the matrices $\boldsymbol{M}$ and $\boldsymbol{K}$ using an overlapping decomposition of the spatial mesh [46, 49].

Let $\mathcal{T}_H$ be a coarse mesh covering Ω with mesh size H, and $\mathcal{T}_h$ be a fine mesh with mesh size h. Denote V_0 and V_h as the finite element spaces consisting of continuous piecewise linear functions associated with the meshes $\mathcal{T}_H$ and $\mathcal{T}_h$, respectively. We introduce a non-overlapping decomposition $\Omega = \sum_{i=1}^{N_p} \Omega_i$ on $\mathcal{T}_h$, where each Ω_i is a union of some elements from $\mathcal{T}_h$, and N_p is the number of subdomains. Then, the overlapping subdomains Ω_i' can be obtained by adding some layers of fine mesh elements from the adjacent subdomains. Denote the finite element subspace on Ω_i' as $V_i = V_h \cap H_0^1(\Omega_i')$ and let N_i be the dimension of V_i, we obtain

$$\boldsymbol{V}_h = (V_h)^k \quad \text{and} \quad \boldsymbol{V}_i = (V_i)^k,\ (i = 0, 1, 2, \ldots, N_p),$$

where k is the block size of the chosen BIM. Define the single time step restriction matrix $\boldsymbol{R}_i \in \mathbb{R}^{N_i\times N} : V_h \to V_i$, the block restriction matrix $\boldsymbol{\mathcal{R}}_i \in \mathbb{R}^{kN_i\times kN} : \boldsymbol{V}_h \to \boldsymbol{V}_i$ can then be defined as

$$\boldsymbol{\mathcal{R}}_i = diag\{\boldsymbol{R}_i, \boldsymbol{R}_i, \cdots, \boldsymbol{R}_i\}.$$

Therefore, the space $\boldsymbol{V}_h$ admits the following decomposition

$$\boldsymbol{V}_h = \sum_{i=1}^{N_p} \boldsymbol{\mathcal{R}}_i^T \boldsymbol{V}_i \qquad \text{and} \qquad \boldsymbol{V}_h = \boldsymbol{\mathcal{R}}_0^T \boldsymbol{V}_0 + \sum_{i=1}^{N_p} \boldsymbol{\mathcal{R}}_i^T \boldsymbol{V}_i, \tag{5.10}$$

which are needed for the one-level and two-level methods, respectively. Here $\boldsymbol{\mathcal{R}}_i^T$ denotes the transpose of $\boldsymbol{\mathcal{R}}_i$ $(i = 0, 1, \ldots, N_p)$. For $\boldsymbol{\mathcal{R}}_0$ there are many choices, in this paper, we only consider these based on the basis functions of the coarse and fine finite element spaces. $\boldsymbol{\mathcal{R}}_i$ $(i > 0)$ is a sub-identity matrix whose diagonal elements corresponding to the subdomain Ω_i are one and all other elements are zero. Note that the coarse and fine meshes don't have to be nested. Then, we can define the following preconditioners.

Table 7: The condition numbers of the matrices with preconditioners $\boldsymbol{\mathcal{P}}^{(1)}$ and $\boldsymbol{\mathcal{P}}^{(2)}$ for Algorithm 3.1 applied to Example 4

k	$\kappa(\boldsymbol{\mathcal{A}})$	$\kappa(\boldsymbol{\mathcal{P}}^{(1)}\boldsymbol{\mathcal{A}})$	$\kappa(\boldsymbol{\mathcal{P}}^{(2)}\boldsymbol{\mathcal{A}})$
2	2.10e+3	122.66	16.43
3	8.76e+3	128.26	17.12
4	1.18e+4	159.73	20.69
5	2.58e+3	198.29	25.77
6	3.29e+3	252.68	39.13
7	4.74e+3	366.23	71.59
8	6.83e+3	531.51	158.74

- One-level additive Schwarz preconditioner

$$\boldsymbol{\mathcal{P}}^{(1)} = \sum_{i=1}^{N_p} \boldsymbol{\mathcal{R}}_i^T \boldsymbol{\mathcal{A}}_i^{-1} \boldsymbol{\mathcal{R}}_i. \tag{5.11}$$

- Two-level additive Schwarz preconditioner

$$\boldsymbol{\mathcal{P}}^{(2)} = \boldsymbol{\mathcal{R}}_0^T \boldsymbol{\mathcal{A}}_0^{-1} \boldsymbol{\mathcal{R}}_0 + \sum_{i=1}^{N_p} \boldsymbol{\mathcal{R}}_i^T \boldsymbol{\mathcal{A}}_i^{-1} \boldsymbol{\mathcal{R}}_i, \tag{5.12}$$

where $\boldsymbol{\mathcal{A}}_0 = \boldsymbol{\mathcal{R}}_0 \boldsymbol{\mathcal{A}} \boldsymbol{\mathcal{R}}_0^T$ is the restriction of $\boldsymbol{\mathcal{A}}$ to the coarse space $\boldsymbol{V}_0$, $\boldsymbol{\mathcal{A}}_i = \boldsymbol{\mathcal{R}}_i \boldsymbol{\mathcal{A}} \boldsymbol{\mathcal{R}}_i^T = \boldsymbol{A} \otimes \boldsymbol{M}_i + \tau \boldsymbol{B} \otimes \boldsymbol{K}_i$ is the restriction of $\boldsymbol{\mathcal{A}}$ to the subspaces $\boldsymbol{V}_i$ $(i \geq 1)$, and the matrices $\boldsymbol{M}_i$ and $\boldsymbol{K}_i$ are defined by

$$\boldsymbol{M}_i = \boldsymbol{R}_i \boldsymbol{M} \boldsymbol{R}_i^T \quad \text{and} \quad \boldsymbol{K}_i = \boldsymbol{R}_i \boldsymbol{K} \boldsymbol{R}_i^T.$$

All inverses in (5.11) and (5.12) are understood as subspace inverse. In practical applications, they are often approximated to save computational cost.

Remark 5.5. We remark that it is important to select the appropriate preconditioner for each practical application. For example if the number of spatial variables is small, such as in ODEs, then the DD method is not necessary. The one-level DD method is useful when the number of processors is small, and the two-level DD methods are for the situations when the number of subdomains is large.

5.3. Some numerical studies of the preconditioners. In this section, we investigate the performance of the proposed Schwarz preconditioners in terms of the condition number, the eigenvalue distribution and the number of GMRES iterations. In the experiments, $\Omega = [0,1] \times [0,1]$ is covered by a uniform coarse mesh of size $H = 1/16$, and a uniform fine mesh of size $h = 1/64$. The spatial fine mesh is decomposed into 4×4 subdomains. The overlapping size is 1 and the time step size τ is chosen as $\tau = h^{2/p}$ ($p = k+1$ when k is odd and $p = k+2$ when k is even), where p denotes the order of accuracy in time. All the preconditioners are constructed exactly; i.e., the inverse of the submatrix and coarse matrix are computed exactly.

EXAMPLE 4. *We consider an advection-diffusion equation*

$$\begin{cases} u_t - \Delta u + u_x + u_y &= f, & \text{in } \Omega \times (0,T], \\ u(x,y,t) &= 0, & \text{on } \partial\Omega \times (0,T], \\ u(x,y,0) &= \sin(\pi x)\sin(\pi y), & \text{in } \Omega, \end{cases}$$

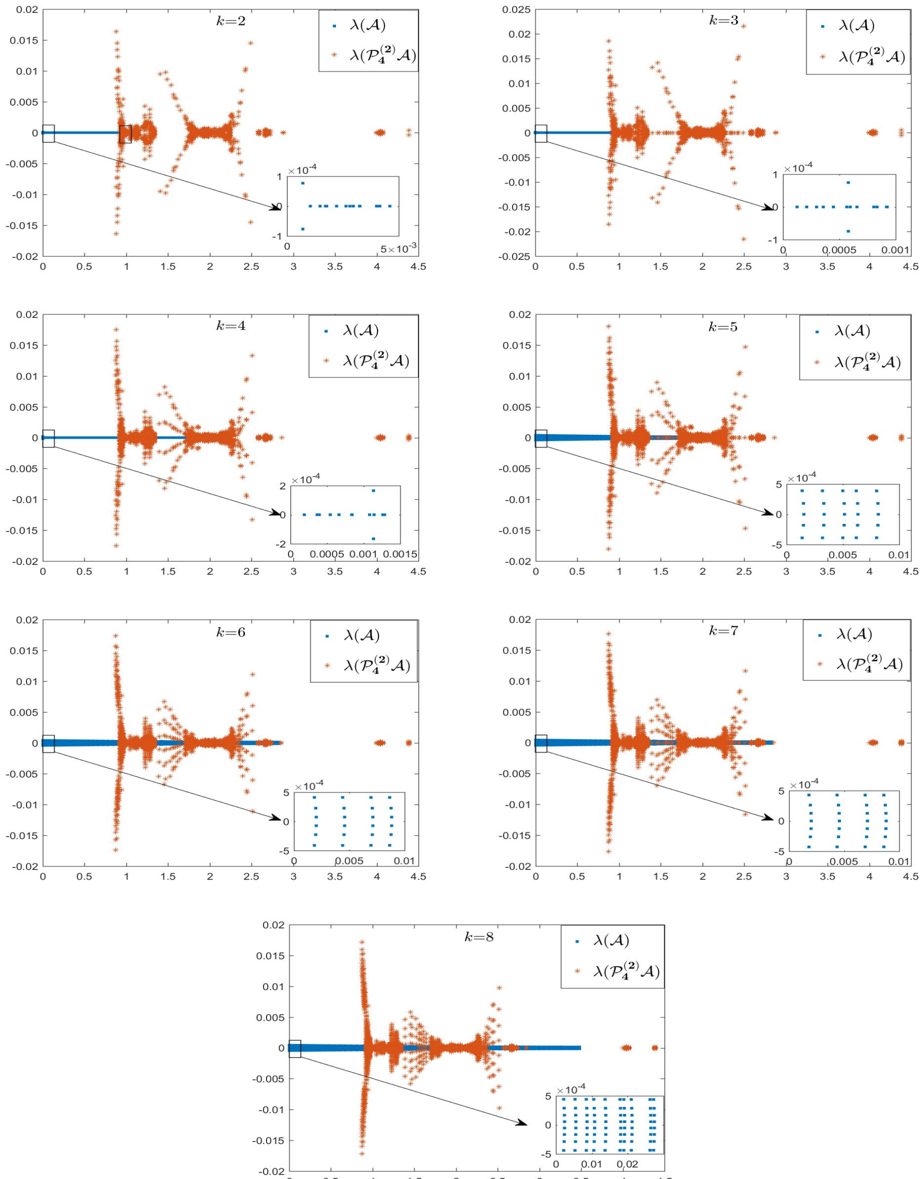

Fig. 6: All eigenvalues of the matrices $\boldsymbol{\mathcal{A}}$ (blue) and $\boldsymbol{\mathcal{P}}^{(2)}\boldsymbol{\mathcal{A}}$ (red) for Algorithm 3.1, zoom-in is for the eigenvalues of $\boldsymbol{\mathcal{A}}$ (near 0)

where f is chosen such that the exact solution is $u(x,y,t) = \sin(\pi x)\sin(\pi y)e^{-t}$.

We study Algorithm 3.1 for Example 4. Table 7 presents the 2-norm condition number of the matrix $\boldsymbol{\mathcal{A}}$ and of $\boldsymbol{\mathcal{A}}$ preconditioned by $\boldsymbol{\mathcal{P}}^{(1)}$ and $\boldsymbol{\mathcal{P}}^{(2)}$, respectively. It shows that the condition number of $\boldsymbol{\mathcal{A}}$ grows firstly with k from 2 to 4, and then there is a drop from $k = 4$ to $k = 5$, afterward, it increases again with k ranging from 5 to 8. The reason is that the BIM matrix $\boldsymbol{B}$ is chosen as a positive diagonal

Table 8: The number of GMRES iterations with the one-level additive Schwarz preconditioner $\mathcal{P}^{(1)}$ for Algorithm 3.1 applied to Example 4 with $h = 1/128$ and $\tau = h^{2/p}$

N_p	16			64			256		
k \ δ	0	1	2	0	1	2	0	1	2
2	55	37	30	72	50	38	94	63	55
3	86	59	47	99	85	65	180	111	98
4	109	70	58	161	110	83	202	143	119
5	118	86	72	161	118	97	226	150	129
6	109	82	70	163	109	94	218	157	130
7	112	83	66	155	111	89	205	145	119
8	112	80	68	155	111	87	200	145	120

Table 9: The number of GMRES iterations with the two-level additive Schwarz preconditioner $\mathcal{P}^{(2)}$ for Algorithm 3.1 applied to Example 4 with $H = 1/16$, $h = 1/128$ and $\tau = h^{2/p}$

N_p	16			64			256		
k \ δ	0	1	2	0	1	2	0	1	2
2	27	23	22	28	24	23	44	33	28
3	27	23	22	28	24	24	44	35	29
4	26	23	22	27	23	23	33	28	25
5	25	21	21	26	23	22	40	32	27
6	24	21	21	25	22	22	40	30	27
7	24	21	20	25	22	21	30	27	23
8	23	21	20	25	22	21	38	30	26

matrix for the cases $k = 2, 3, 4$ and as an identity matrix for the cases $k = 5, 6, 7, 8$. It seems that the preconditioner with identity matrix $\boldsymbol{B}$ yields a lower condition number than the preconditioner with the positive diagonal matrix $\boldsymbol{B}$. All the preconditioners are able to reduce the condition number, which increases with k, and the two-level preconditioner $\mathcal{P}^{(2)}$ performs better than the one-level preconditioner $\mathcal{P}^{(1)}$.

When the Schwarz preconditioners are accelerated by GMRES [46, 49], the eigenvalue distribution of the preconditioned matrix is an important indicator for the performance of GMRES. In Figure 6, we plot the eigenvalues of $\mathcal{A}$ and the preconditioned matrix $\mathcal{P}^{(2)}\mathcal{A}$ for Algorithms 3.1. It is clear that some eigenvalues of $\mathcal{A}$ are close to the origin and the eigenvalues of $\mathcal{P}^{(2)}\mathcal{A}$ are all away from the origin. Moreover, it is interesting that the eigenvalues of $\mathcal{A}$ are clustered on a single line for BIM with a positive diagonal matrix $\boldsymbol{B}$ and are clustered on k lines for BIM with an identity matrix $\boldsymbol{B}$.

Finally, we solve Example 4 using GMRES preconditioned by $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$. In the implementation, GMRES(30) is used with relative tolerance 10^{-6}. In Table 8, we report iteration counts for $\mathcal{P}^{(1)}$ by varying the number of subdomians N_p, overlapping size δ and block size k, the spacial mesh size and time step size are $h = 1/128$ and $\tau = h^{2/p}$, respectively. For a given block size k, it shows the iteration counts decrease with the increase of the overlapping size, and grow with the number of subdomains. Moreover, the iteration counts increase a little from $k = 2$ to $k = 4$, and changes slightly from $k = 4$ to $k = 8$. For the two-level preconditioner $\boldsymbol{\mathcal{P}}^{(2)}$, the coarse and fine mesh sizes are $H = 1/16$ and $h = 1/128$, respectively. The results in Table 9 show that the iteration counts decrease with the overlapping size and is bounded independently of the number of subdomains. Further, the number of iterations doesn't change much with the block size k.

6. Conclusions. In this paper we developed a unified framework for the class of BIM with A-stability for parabolic problems. Similar to IRK, BIM offers high order of accuracy, desirable stability properties, and requires a single initial value. Because of the flexibility in selecting the BIM matrices, the resulting large, highly ill-conditioned linear system is easier to precondition and solve than that from IRK. For some block sizes, we derived the matrix form of $\boldsymbol{B}^{-1}\boldsymbol{A}$ explicitly, and show that it is positive stable, moreover, we also derived positive diagonal matrices $\boldsymbol{B}$ and positive definite matrices $\boldsymbol{A}$ explicitly. The other important result is that, the positive definiteness of the coupled matrix $\boldsymbol{\mathcal{A}}$ depends only on the positive definiteness of the spatial matrix $\boldsymbol{K}$. Using the properties of the BIM matrices, we developed a finite element theory which is not possible for IRK because the lack of the positive definiteness of the Runge-Kutta matrix. We also introduced and studied numerically some DD preconditioners that are quite effective for the system of equations arising from the BIM discretization.

Appendix A. Coefficients of the stability functions for BIM.

Table 10: Coefficients of the stability functions for BIM($2 \leq k \leq 8$)

		z^8	z^7	z^6	z^5	z^4	z^3	z^2	z	1
$k=2$	$\lvert C_1(z)\rvert$							-1	0	6
	$\lvert C_2(z)\rvert$							2	6	6
	$\lvert C(z)\rvert$							2	-6	6
$k=3$	$\lvert C_1(z)\rvert$						1	-1	-6	12
	$\lvert C_2(z)\rvert$						-1	-1	6	12
	$\lvert C_3(z)\rvert$						3	11	18	12
	$\lvert C(z)\rvert$						-3	11	-18	12
$k=4$	$\lvert C_1(z)\rvert$					-3	5	15	-60	60
	$\lvert C_2(z)\rvert$					2	0	-15	0	60
	$\lvert C_3(z)\rvert$					-3	-5	15	60	60
	$\lvert C_4(z)\rvert$					12	50	105	120	60
	$\lvert C(z)\rvert$					12	-50	105	-120	60
$k=5$	$\lvert C_1(z)\rvert$				12	-26	-45	300	-540	360
	$\lvert C_2(z)\rvert$				-6	4	45	-60	-180	360
	$\lvert C_3(z)\rvert$				6	4	-45	-60	180	360
	$\lvert C_4(z)\rvert$				-12	-26	45	300	540	360
	$\lvert C_5(z)\rvert$				60	274	675	1020	900	360
	$\lvert C(z)\rvert$				-60	274	-675	1020	-900	360
$k=6$	$\lvert C_1(z)\rvert$			-60	154	147	-1680	4200	-5040	2520
	$\lvert C_2(z)\rvert$			24	-28	-168	420	420	-2520	2520
	$\lvert C_3(z)\rvert$			-18	0	147	0	-840	0	2520
	$\lvert C_4(z)\rvert$			24	28	-168	-420	420	2520	2520
	$\lvert C_5(z)\rvert$			-60	-154	147	1680	4200	5040	2520
	$\lvert C_6(z)\rvert$			360	1764	4872	8820	10500	7560	2520
	$\lvert C(z)\rvert$			360	-1764	4872	-8820	10500	-7560	2520
$k=7$	$\lvert C_1(z)\rvert$		90	-261	-105	2667	-8400	13860	-12600	5040
	$\lvert C_2(z)\rvert$		-30	47	189	-693	0	3780	-7560	5040
	$\lvert C_3(z)\rvert$		18	-9	-147	-147	840	-1260	-2520	5040
	$\lvert C_4(z)\rvert$		-18	-9	147	147	-840	-1260	2520	5040
	$\lvert C_5(z)\rvert$		30	47	-189	-693	0	3780	7560	5040
	$\lvert C_6(z)\rvert$		-90	-261	105	2667	8400	13860	12600	5040
	$\lvert C_7(z)\rvert$		630	3267	9849	20307	29400	28980	17640	5040
	$\lvert C(z)\rvert$		-630	3267	-9849	20307	-29400	28980	-17640	5040
$k=8$	$\lvert C_1(z)\rvert$	-210	669	-16	-6363	24045	-49140	61740	-45360	15120
	$\lvert C_2(z)\rvert$	60	-114	-331	1638	-1155	-7560	23940	-30240	15120
	$\lvert C_3(z)\rvert$	-30	27	236	-441	-1155	3780	1260	-15120	15120
	$\lvert C_4(z)\rvert$	24	0	-205	0	1365	0	-6300	0	15120
	$\lvert C_5(z)\rvert$	-30	-27	236	441	-1155	-3780	1260	15120	15120
	$\lvert C_6(z)\rvert$	60	114	-331	-1368	-1155	7560	23940	30240	15120
	$\lvert C_7(z)\rvert$	-210	-669	-16	6363	24045	49140	61740	45360	15120
	$\lvert C_8(z)\rvert$	1680	9132	29531	67284	112245	136080	114660	60480	15120
	$\lvert C(z)\rvert$	1680	-9132	29531	-67284	112245	-136080	114660	-60480	15120

REFERENCES

[1] L. Aceto and D. Trigiante, *On the A-stable methods in the GBDF class*, Nonlinear Anal.: Real World Appl. 3 (2002), pp. 9-23.

[2] O. A. Akinfenwa, S. N. Jator, and N. M. Yao, *Continuous block backward differentiation formula for solving stiff ordinary differential equations*, Comput. Math. Appl. 65 (2013), pp. 996-1005.

[3] G. P. Barker, A. Berman, and R. J. Plemmons, *Positive diagonal solutions to the Lyapunov equations*, Linear and Multilinear Algebra, 5 (1978), pp. 249-256.

[4] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia, 1994.

[5] L. Brugnano, *Blended block BVMs (B3VMs): A family of economical implicit methods for ODEs*, J. Comput. Appl. Math. 116 (2000), pp. 41-62.
[6] L. Brugnano and C. Magherini, *Blended implementation of block implicit methods for ODEs*, Appl. Numer. Math. 42 (2002), pp. 29-45.
[7] L. Brugnano and D. Trigiante, *Block implicit methods for ODEs*, Recent trends in numerical analysis, pp. 81-105, Adv. Theory Comput. Math., 3, Nova Sci. Publ., Huntington, NY, 2001.
[8] L. Brugnano and D. Trigiante, *Solving Differential Problems by Multistep Initial and Boundary Value Methods*, Gordon and Breach, Amsterdam, 1998.
[9] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, John Wiley & Sons, Ltd., Chichester, 2016.
[10] X.-C. Cai, *Additive Schwarz algorithms for parabolic convection-diffusion equations*, Numer. Math., 60 (1991), pp. 41-61.
[11] X.-C. Cai, *Multiplicative Schwarz methods for parabolic problems*, SIAM J. Sci. Comput., 15 (1994), pp. 587-603.
[12] W. Chen, W. Li, C. Wang, S. Wang, and X. Wang, *Energy stable higher-order linear ETD multi-step methods for gradient flows: application to thin film epitaxy*, Res. Math. Sci., 7 (2020).
[13] K. Cheng, Z. Qiao, and C. Wang, *A third order exponential time differencing numerical scheme for no-slope-selection epitaxial thin Film model with energy stability*, J Sci. Comput., 81 (2019), pp. 154–185.
[14] K. Cheng, C. Wang, S. M. Wise, and Y. Wu, *A third order accurate in time, BDF-type energy stable scheme for the Cahn-Hilliard equation*, Numer. Math. Theory Methods Appl., 15 (2022), pp. 279–303.
[15] C. Cong, X.-C. Cai, and K. Gustafson, *Implicit space-time domain decomposition methods for stochastic parabolic partial differential equations*, SIAM J. Sci. Comput., 36 (2014), pp. C1-C24.
[16] W. Coppet, *Stability and Asymptotic Behavior of Differential Equations*, Heath, Boston, 1965.
[17] M. Crouzeix and F. Ruamps, *On rational approximations to the exponential*. RAIRO Anal. Numér., 11 (1977), pp. 241-243.
[18] G. Dahlquist, *A special stability problem for linear multistep methods*, BIT, 3 (1963), pp. 27-43.
[19] Q. Du, L. Ju, X. Li, and Z. Qiao, *Maximum bound principles for a class of semilinear parabolic equations and exponential time-differencing schemes*, SIAM Rev. 63 (2021), pp. 317-359.
[20] J. C. Geromel, *On the determination of a diagonal solution of the Lyapunov equation*, IEEE Trans. Automat. Control. 30 (1985), pp. 404-486.
[21] S. Gottlieb and D. I. Ketcheson, *Time discretization techniques*, Handbook of numerical methods for hyperbolic problems, 549-583, Handb. Numer. Anal., 17, Elsevier/North-Holland, Amsterdam, 2016.
[22] Y. Hao, Q. Huang, and C. Wang, *A third order BDF energy stable linear scheme for the no-slope-selection thin film model*, Commun. Comput. Phys., 29 (2021). pp. 905-929.
[23] E. Hairer, S. P. NØsett and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer, Berlin, 1993.
[24] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations. II: Stiff and Differential-Algebraic Problems*, Springer, Berlin, 1996.
[25] D. Hershkowitz and H. Schneider, *Scalings of vector spaces and the uniqueness of Lyapunov scaling factors*, Linear Multilinear Algebra, 17 (1985), pp. 203-226.
[26] M. Hochbruck and A. Ostermann, *Exponential integrators*, Acta Numer. 19 (2010), pp. 209-286.
[27] H. Hu, *A projective method for rescaling a diagonally stable matrix to be positive definite*, SIAM J. Matrix Anal. Appl. 13 (1992), pp. 1255-1263.
[28] L. O. Jay, *Inexact simplified Newton iterations for implicit Runge-Kutta methods*, SIAM J. Numer. Anal. 38 (2000), pp. 1369-1388.
[29] L. O. Jay and T. Braconnier, *A parallelizable preconditioner for the iterative solution of implicit Runge-Kutta type methods*, J. Comput. Appl. Math. 111 (1999), pp. 63-76.
[30] X. Jiao, X. Wang, and Q. Chen, *Optimal and low-memory near-optimal preconditioning of fully implicit Runge-Kutta schemes for parabolic PDEs*, SIAM J. Sci. Comput. 43 (2021), pp. A3527-A3551.
[31] E. Kaszkurewicz and A. Bhaya, *Matrix Diagonal Stability in Systems and Computation*, Springer, 2000.
[32] C. A. Kennedy and M. H. Carpenter, *Diagonally implicit Runge-Kutta methods for ordinary differential equations. A review*, Langley Research Center, Hamp-ton, VA, 2016, NASA

Technical Memorandum.
[33] H. K. Khalil, *On the existence of positive diagonal P such that $PA + A'P < 0$*, IEEE Trans. Automat. Control, 27 (1982), pp. 181-184.
[34] O. Y. Kushel, *Unifying matrix stability concepts with a view to applications*, SIAM Rev. 61 (2019), pp. 643-729.
[35] J. Van lent and S. Vandewalle, *Multigrid methods for implicit Runge-Kutta and boundary value method discretizations of PDEs*, SIAM J. Sci. Comput. 27 (2005), pp. 67-92.
[36] S. Li and X-C. Cai, *Convergence analysis of two-level space-time additive Schwarz method for parabolic equations*, SIAM J. Numer. Anal. 53 (2015), pp. 2727-2751.
[37] S. Li, X. Shao, and X.-C. Cai, *Multilevel space-time additive Schwarz methods for parabolic equation*, SIAM J. Sci. Comput. 40 (2018), pp. A3012-A3037.
[38] S. Li, X. Shao, and R. Chen, *Multilevel space-time multiplicative Schwarz preconditioner for parabolic equations*, Numer. Linear Algebra Appl. 28 (2021), e2390.
[39] A. Lyapunov, *Problème générale de la stabilité: du mouvement*, Communicutiuns of the Math. Sot. of Khurkow, 1892, republ. in Ann. Math. Stud., 17, Princeton Press, Princeton, 1947.
[40] K.-A. Mardal, T. K. Nilssen, and G. A. Staff, *Order-optimal preconditioners for implicit Runge-Kutta schemes applied to parabolic PDEs*, SIAM J. Sci. Comput. 29 (2007), 361-375.
[41] W. E. Milne, *Numerical Solution of Differential Equations*, Wiley, New York, 1953.
[42] H. Ramos and P. Popescu, *How many k-step linear block methods exist and which of them is the most efficient and simplest one?*, Appl. Math. Comput. 316 (2018), pp. 296-309.
[43] Md. M. Rana, V. E. Howle, K. Long, A. Meek, and W. Milestone, *A new block preconditioner for implicit Runge-Kutta methods for parabolic PDE problems*, SIAM J. Sci. Comput. 43 (2021), pp. S475-S495.
[44] J. B. Rosser, *A Runge-Kutta for all seasons*, SIAM Rev. 9 (1967), pp. 417-452.
[45] L. F. Shampine and H. A. Watts, *Block implicit one-step methods*, Math. Comp. 23 (1969), pp. 731-740.
[46] B. Smith, P. Bjørstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.
[47] G. A. Staff, K.-A. Mardal, and T. K. Nilssen, *Preconditioning of fully implicit Runge-Kutta schemes for parabolic PDEs*. Model. Identif. Control. 27 (2006), pp. 109-123.
[48] V. Thomée, *Galerkin Finite Element Methods for Parabolic Problems*. Second Edition, Springer-Verlag, Berlin, 2006.
[49] A. Toselli and O. B. Widlund, *Domain Decomposition Methods-Algorithms and Theory*, Springer-Verlag Berlin Heidelberg, 2005.
[50] G. Wanner and E. Hairer, *Solving Ordinary Differential Equations II. Stiffand Differential-Algebraic Problems*, Springer-Verlag, Berlin, Heidelberg, 1991.
[51] H. A. Watts, *A Stable Block Implicit One-Step Methods*, PhD Thesis, The University of New Mexico. 1971.
[52] H. A. Watts and L. F. Shampine, *A-stable block one-step methods*, BIT. 12 (1972), pp. 252-266.
[53] O. B. Widlund, *A note on unconditionally stable linear multistep methods*, BIT. 7 (1967), pp. 65-70.