ORDER–OPTIMAL PRECONDITIONERS FOR IMPLICIT RUNGE–KUTTA SCHEMES APPLIED TO PARABOLIC PDES

K.–A. MARDAL, T. K. NILSSEN, AND G. A. STAFF

Abstract. In this paper we show that standard preconditioners for parabolic PDEs discretized by implicit Euler or Crank–Nicolson schemes can be reused for higher–order fully implicit Runge–Kutta time discretization schemes. We prove that the suggested block diagonal preconditioners are order–optimal for A–stable and irreducible Runge–Kutta schemes with invertible coefficient matrices. The theoretical investigations are confirmed by numerical experiments.

1. Introduction

Let Ω be a bounded polygonal domain in $\mathbb{R}^d$, with d=2 or 3, and boundary $\partial\Omega$. In this paper we will consider a parabolic PDE on the form

$$\frac{\partial u}{\partial t} = \Delta u + f, \quad \text{in } \Omega,\ t > 0, \tag{1}$$

$$u = 0, \quad \text{on } \partial\Omega,\ t \geq 0 \tag{2}$$

$$u = u^0, \quad \text{in } \Omega,\ t = 0. \tag{3}$$

The equations (1)–(3) are discretized in space, by a finite element method, which gives a system of ODEs to be solved,

$$M\frac{du_h}{dt} = -Ku_h + f_h, \quad t > 0, \tag{4}$$

$$u_h = u_h^0, \quad t = 0, \tag{5}$$

where M is the mass matrix and K is the stiffness matrix. Higher–order time discretization schemes based on fully implicit Runge–Kutta schemes are rarely used when discretizing (4)–(5), due to the additional cost of solving a larger and more complicated system of equations. The lack of efficient solution algorithms for such systems has prevented these algorithms from widespread use for PDEs, despite the many appealing properties of such schemes. However, the lack of efficient algorithms does not only apply to PDEs, it applies to ODEs in general. A fact we emphasize by quoting some standard ODE textbooks:
1) "*An efficient solution of this system is the main problem in the implementation of an implicit Runge–Kutta method*" [8, p. 118].
2) "*One of the challenges for implicit Runge–Kutta methods is the development of efficient implementations.... Thus, it is important to look for ways to make the iterations process less expensive*" [2, p. 103].
In this paper we will see how one can reuse fast solution algorithms developed for low order time-discretizations, in the case of a parabolic PDE.

When the equations (4)–(5) are discretized by lower order time discretization schemes such as implicit Euler, we arrive at the following sequence of linear systems to be solved at each time step,

$$(M + \delta t K)u_h^n = Mu^{n-1} + \delta t f_h^n, \tag{6}$$

where δt is the time stepping parameter. Order optimal solution algorithms for this system have been found for most spatial discretization methods. The goal in this

paper is to reuse such preconditioners for fully implicit Runge–Kutta schemes. In that way the method will have implementational similarities with BDF and DIRK methods, but the stability and convergence of the chosen fully implicit Runge–Kutta method. Furthermore, we will prove that these preconditioners are order–optimal with respect to δt and h and demonstrate their efficiency by several numerical experiments. Order-optimality in s, where s is the number of stage values in the Runge-Kutta schemes, is not obtained. However, we demonstrate that the s-dependency is weak compared to the accuracy gained. Furthermore, we have in [19] numerically demonstrated that this s-dependency can be reduced by generalising a Gauss-Seidel preconditioner with certain coefficients determined by a simple optimization principle.

Except for BDF and DIRK methods, which only require the solution of systems on the form (6), the authors only know of a few other works on solution algorithms for higher–order time discretization schemes in the context of PDEs [5], [7], [11], [12], [13], [14], and [15]. In [12] and [13] they present an inexact block LU preconditioners when using the W-transformation on the Runge-Kutta schemes. Block Jacobi and Gauss-Seidel preconditioners for generalized Adams methods are studied in [11]. In [15], we discussed preconditioners for higher–order Padé formulations of parabolic PDEs where the matrix was a polynomial of $(M + \delta tK)$-like matrices, on the form $\Pi_{i=1}^{n}(M + a_i\delta tK)^i$. In this work, we instead consider linear systems of $(M + \delta tK)$-like matrices.

In the recent work [14], multigrid methods for Runge–Kutta methods was studied. A key idea in this work was to employ block smoothers, simultaneously updating all unknowns related to a spatial grid point. Convergence rate independent of the number of quadrature nodes of the Runge–Kutta scheme, s, was obtained, but the smoothers require the solution of $s \times s$ systems at each spatial grid point. Our approach is different in the sense that we employ standard preconditioners (e.g., multigrid with pointwise smoothers or domain decomposition) for (6) on each of the blocks in the block matrix arising from the Runge–Kutta discretization.

The order-optimality of the method proposed here is proven by viewing the differential operator as an isomorphism in appropriate spaces. This gives an upper bound on the condition number of the preconditioned system, and therefore an upper bound on the number of iterations. Other works taking the same approach are [1], [9], and [17].

The preconditioners discussed in this paper can be (and have been) implemented as a block diagonal matrix of standard preconditioners. Practical details concerning the implementation of block preconditioners (in the C++ library Diffpack) can be found in [16]. More numerical experiments can be found in [19]. Here, numerical experiments also in 3D verify the order-optimality. Furthermore, it is shown numerically that block Gauss-Seidel preconditioners can improve the efficiency of the preconditioner.

The remaining part of the paper is organized as follows: In Section 2 we review some useful concepts. In Section 3 we prove that the continuous version of the time stepping operator for Runge–Kutta methods is an isomorphism bounded independent of δt, when using the proper interpolation spaces. This enables us to construct order–optimal preconditioners in the discrete case in Section 4. Section 5 presents some numerical experiments.

2. Preliminaries

Let $H^1 = H^1(\Omega)$ be the Sobolev space with first order derivatives in $L^2 = L^2(\Omega)$, with norm $\|\cdot\|_1$. The space H_0^1 is the closure of C_0^∞ in H^1, with norm $\|\cdot\|_{H_0^1} = |\cdot|_1$, which is the L_2 norm of the first order derivatives. The $|\cdot|_1$ norm is equivalent

with the standard H^1 norm on H_0^1. The space H^{-1} is the dual space of H_0^1. Vector valued functions and spaces are denoted by bold face symbols. The inner product of both scalars, vectors, and matrices, as well as the duality pairing of H_0^1 and H^{-1} and their corresponding vector spaces, are denoted by $(\cdot,\cdot)$, and the corresponding norm is denoted by $\|\cdot\|$. The operators Δ and ∇ are applied to both scalars and vectors, the scalar versions are standard and the vector versions are defined,

$$(\Delta \boldsymbol{u})_i = \Delta u_i, \quad \text{and } (\nabla \boldsymbol{u})_{i,j} = \frac{\partial u_i}{\partial x_j}, \quad \text{for } 1 \le i \le s,\ 1 \le j \le d,$$

where u_i is the i'th component of $\boldsymbol{u}$. Notice that $\nabla \boldsymbol{u}$ is a $s \times d$ matrix. The spaces involving time are defined by

$$\|u\|_{L^2(0,T;X)} = (\int_0^T \|u(t)\|_X^2 dt)^{1/2},$$

where $\|\cdot\|_X$ is the spatial norm in the space X (which is either, H_0^1, H^{-1} or L^2). The weak form of (1)–(3) is:
Find $u \in L^2(0,T;H_0^1)$ with $\frac{\partial u}{\partial t} \in L^2(0,T;H^{-1})$ such that

$$(\frac{\partial u}{\partial t}, v) + (\nabla u, \nabla v) = (f, v), \quad \forall v \in H_0^1, t > 0, \tag{7}$$

$$u = u_0, \quad t = 0. \tag{8}$$

Similarly, the finite element formulation is defined by seeking an approximation $u_h(t)$ in a finite element subspace $V_h \subset H_0^1$ by:
Find $u_h \in L^2(0,T;V_h)$ with $\frac{\partial u_h}{\partial t} \in L^2(0,T;V_h)$ such that

$$\begin{aligned}(\frac{\partial u_h}{\partial t}, v) + (\nabla u_h, \nabla v) &= (f, v), \quad \forall v \in V_h, t > 0,\\ u &= u_0, \quad t = 0.\end{aligned}$$

This can be written as a linear system of ODEs,

$$I_h \frac{\partial u_h}{\partial t} - \Delta_h u_h = f_h,$$

where I_h is the identity operator (the mass matrix M) and Δ_h is the Δ operator (the negative stiffness matrix, $-K$) on V_h. The right-hand side f_h is the L_2-projection of f on V_h (often approximated by the mass matrix times a vector with the values of f evaluated at nodal points). We may write this system as

$$\frac{\partial u_h}{\partial t} - I_h^{-1}\Delta_h u_h = I_h^{-1} f_h,$$

and the eigenvalues of $-I_h^{-1}\Delta_h$ are real and positive.

The unknown $u_h(t)$ is approximated by a Runge–Kutta method,

$$u_h^n = u_h^{n-1} + \delta t \sum_{i=1}^{s} b_i u_{h,i}^n,$$

where

$$I_h u_{h,i}^n = \Delta_h(u_h^{n-1} + \delta t \sum_{j=1}^{s} a_{ij} u_{h,j}^n) + f_h(t_{n-1} + c_i \delta t), \quad 1 \le i \le s, \tag{9}$$

where $u_{h,i}^n$ is the intermediate stage value approximating the time derivative of u at a given quadrature node (we use the slightly confusing variable name $u_{h,i}^n$ for the time derivatives of u, since $u_{h,i}^n$ is the unknown that appears in our linear system). This system can be written as,

$$(\boldsymbol{I}^{s,s} \otimes I_h - \delta t A \otimes \Delta_h)\boldsymbol{u}_h^n = -\boldsymbol{I}^{s,s} \otimes \Delta_h \boldsymbol{u}_h^{n-1} + \boldsymbol{f}_h^n,$$

where $\boldsymbol{f}_h^n = (f_{h,1}^n, \dots f_{h,s}^n)^T$, $f_{h,i}^n = f_h(t_{n-1} + c_i\delta t)$, $\boldsymbol{u}_h^n$ is the vector of intermediate stage values $\boldsymbol{u}_h^n = (u_{h,1}^n, \dots, u_{h,s}^n)^T$, $\boldsymbol{I}^{s,s}$ is the identity matrix in $\mathbb{R}^{s,s}$, A, b and c are the Runge–Kutta coefficients, weights and nodes, and $\otimes$ is the Kronecker tensor product,

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{n1}B & \dots & a_{nn}B \end{pmatrix}.$$

A Runge–Kutta scheme is A–stable if the stability function,

$$R(z) = 1 + zb^T(I - zA)^{-1}\mathbf{1}$$

satisfies $|R(z)| \leq 1$ when $\mathrm{Re}(z) < 0$, where $\mathbf{1} \in \mathbb{R}^s$ is a vector of ones. In this paper we will assume that the Runge–Kutta methods are A–stable, irreducible and has invertible A, such as Gauss, LobattoIIIC, RadauIA, and RadauIIA, cf. e.g. [8]. We will also assume that the diagonal entries of A are positive. We refer to [6] for a description of parabolic PDEs and [20] for corresponding discretization methods.

We use the common definition of an order optimal preconditioner, which is that $\mathcal{B}_k$ is an order optimal preconditioner for $\mathcal{A}_k$, with respect to the parameter k, given that the condition number of $\mathcal{B}_k\mathcal{A}_k$ is bounded independently of the parameter k, and that the evaluation and storage of $\mathcal{B}_k$ is similar to that of $\mathcal{A}_k$. In this paper we will show that the considered preconditioners are order-optimal both with respect to the parameter h and δt.

Finally, we will need the intersection and sum of Hilbert spaces. If X and Y are Hilbert spaces, both continuously contained in some larger Hilbert space, then the intersection $X \cap Y$ and the sum $X + Y$ are Hilbert spaces, and the norms are defined,

$$\|z\|_{X\cap Y} = (\|z\|_X^2 + \|z\|_Y^2)^{1/2}$$

and

$$\|z\|_{X+Y} = \inf_{\substack{z=x+y \\ x\in X,\, y\in Y}} (\|x\|_X^2 + \|y\|_Y^2)^{1/2}.$$

Furthermore, if $X \cap Y$ is dense in both X and Y, then $(X \cap Y)^* = X^* + Y^*$. We refer to [4, Chapter 2] for these results. The spaces we will use are $L^2 \cap \delta t H_0^1$ and its dual space $L^2 + \frac{1}{\delta t}H^{-1}$, and the vector space of these spaces, $\boldsymbol{L}^2 \cap \delta t \boldsymbol{H}_0^1$ and $\boldsymbol{L}^2 + \frac{1}{\delta t}\boldsymbol{H}^{-1}$, with dimension s. We remark that for $\delta t > 0$, $L^2 \cap \delta t H_0^1$ is equal to H^1 as a set, but as $\delta t \to 0$, the norm degenerates to L^2. Since the spaces will appear many times we have adopted the short-hand notation:

$$\begin{aligned} V &= L^2 \cap \delta t H_0^1, \\ V^* &= L^2 + \frac{1}{\delta t}H^{-1}, \\ \boldsymbol{V} &= \boldsymbol{L}^2 \cap \delta t \boldsymbol{H}_0^1, \\ \boldsymbol{V}^* &= \boldsymbol{L}^2 + \frac{1}{\delta t}\boldsymbol{H}^{-1}. \end{aligned}$$

3. An Existence Result

The purpose of this paper is to introduce preconditioners for Runge–Kutta schemes applied to parabolic PDEs that are uniform in both space and time (but not the number of stages). To do this we will employ the spaces introduced in the previous section to describe the spatial problem to be solved at each time step.

We start with an implicit Euler time discretization of (1) to explain the problem: Find $u^n \in V$, such that

$$a^E(u^n, v) = b^n(v), \quad \forall v \in V,$$

where

$$\begin{aligned} a^E(u^n, v) &= (u^n, v) + \delta t(\nabla u^n, \nabla v), \\ b^n(v) &= (u^{n-1}, v) + \delta t(f^n, v). \end{aligned}$$

It then follows that

$$\begin{aligned} a^E(u, v) &\leq c_1^E \|u\|_V \|v\|_V, \\ a^E(u, u) &\geq \frac{1}{c_2^E} \|u\|_V^2. \end{aligned}$$

From the Lax-Milgram theorem (or the Riez representation theorem since a^E is symmetric) we know that there exists a unique solution in V, depending continuously on b^n in V^*. Hence, if we let $\mathcal{A}^E$ be defined by

$$(\mathcal{A}^E u, v) = a^E(u, v)$$

where,

$$\mathcal{A}^E = I - \delta t \Delta,$$

then $\mathcal{A}^E$ is an isomorphism mapping V to V^* bounded independently of δt, i.e.,

$$\|\mathcal{A}^E\|_{\mathcal{L}(V,V^*)} \leq c_1^E \quad \text{and} \quad \|(\mathcal{A}^E)^{-1}\|_{\mathcal{L}(V^*,V)} \leq c_2^E.$$

In contrast, if we had considered $\mathcal{A}^E$ as a mapping from H_0^1 to H^{-1}, then c_1^E and c_2^E would depend on δt.

Similarly, the implicit Runge–Kutta method (i.e. the continuous version of (9)) can be written on the form:
Find $\boldsymbol{u}^n \in \boldsymbol{V}$, such that

$$(\boldsymbol{u}^n, \boldsymbol{v}) + \delta t(A \otimes \Delta \boldsymbol{u}^n, \boldsymbol{v}) = (-\boldsymbol{I}^{s,s} \Delta \boldsymbol{u}^{n-1}, \boldsymbol{v}) + \delta t(\boldsymbol{f}^n, \boldsymbol{v}), \quad \forall \boldsymbol{v} \in \boldsymbol{V}. \tag{10}$$

The solution u^n is obtain by

$$u^{n+1} = u^n + \delta t \sum_{i=1}^{s} b_i u_i^n.$$

Equation (10) may be written as

$$\mathcal{A}\boldsymbol{u} = \boldsymbol{f},$$

where

$$\mathcal{A} = I - \delta t A \otimes \Delta.$$

In this section we will show that $\mathcal{A} : \boldsymbol{V} \to \boldsymbol{V}^*$ is an isomorphism,

$$\|\mathcal{A}\|_{\mathcal{L}(\boldsymbol{V},\boldsymbol{V}^*)} \leq c_1 \quad \text{and} \quad \|\mathcal{A}^{-1}\|_{\mathcal{L}(\boldsymbol{V}^*,\boldsymbol{V})} \leq c_2, \tag{11}$$

where both c_2 and c_1 are independent of δt. The associated bilinear form is

$$\begin{aligned} a(\boldsymbol{u}, \boldsymbol{v}) &= (\mathcal{A}\boldsymbol{u}, \boldsymbol{v}) = ((I - \delta t A \otimes \Delta)\boldsymbol{u}, \boldsymbol{v}) \\ &= (\boldsymbol{u}, \boldsymbol{v}) - \delta t(A\Delta \boldsymbol{u}, \boldsymbol{v}) \\ &= (\boldsymbol{u}, \boldsymbol{v}) + \delta t(A\nabla \boldsymbol{u}, \nabla \boldsymbol{v}). \end{aligned}$$

We remark that $a(\cdot, \cdot)$ is neither positive nor symmetric, for a general A arising from a Runge–Kutta scheme. We can therefore not use the theorems of Riez or Lax–Milgram. To be able to prove that $\mathcal{A}$ is an isomorphism we need the theorem of Babuska and Aziz [3], which in our setting reads as follows.

Theorem 3.1. *$\mathcal{A}$ is an isomorphism mapping $\boldsymbol{V}$ to $\boldsymbol{V}^*$ given that the following conditions are satisfied:*
There exists a c_1 independent of δt such that (boundedness)

$$|a(\boldsymbol{u}, \boldsymbol{v})| \leq c_1 \|\boldsymbol{u}\|_{\boldsymbol{V}} \|\boldsymbol{v}\|_{\boldsymbol{V}}, \quad \forall \boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{V}. \tag{12}$$

There exists a c_2 independent of δt such that (inf–sup)

$$\sup_{\boldsymbol{v}\in\boldsymbol{V}} \frac{a(\boldsymbol{u},\boldsymbol{v})}{\|\boldsymbol{v}\|_{\boldsymbol{V}}} \geq \frac{1}{c_2}\|\boldsymbol{u}\|_{\boldsymbol{V}}, \quad \forall \boldsymbol{u}\in\boldsymbol{V}. \tag{13}$$

For $\boldsymbol{v}\in\boldsymbol{V}$ there exist $\boldsymbol{u}\in\boldsymbol{V}$ such that

$$a(\boldsymbol{u},\boldsymbol{v}) \neq 0. \tag{14}$$

Proof. See [3]. □

Hence, it remains to prove (12)–(14). The first two properties are proven in the following lemmas.

Lemma 3.1. *There exists a constant c_1 independent of δt such that*

$$|a(\boldsymbol{u},\boldsymbol{v})| \leq c_1\|\boldsymbol{u}\|_{\boldsymbol{V}}\|\boldsymbol{v}\|_{\boldsymbol{V}}.$$

Proof. To show (12) we see that

$$\begin{aligned}
a(\boldsymbol{u},\boldsymbol{v}) &= ((I-\delta t A\otimes\Delta)\boldsymbol{u},\boldsymbol{v}) \\
&= (\boldsymbol{u},\boldsymbol{v}) - \delta t \sum_{i,j\leq s} a_{ij}\,(\Delta u_i, v_j) \\
&= (\boldsymbol{u},\boldsymbol{v}) + \delta t \sum_{i,j\leq s} a_{ij}\,(\nabla u_i, \nabla v_j) \\
&\leq a_{max}\left(\|\boldsymbol{u}\|\|\boldsymbol{v}\| + \delta t \sum_{i,j\leq s}\|\nabla u_i\|\|\nabla v_j\|\right) && (15)\\
&\leq c_1\left(\|\boldsymbol{u}\|\|\boldsymbol{v}\| + \delta t\|\nabla\boldsymbol{u}\|\|\nabla\boldsymbol{v}\|\right) && (16)\\
&\leq c_1\left(\|\boldsymbol{u}\|^2 + \delta t\|\nabla\boldsymbol{u}\|^2\right)^{\frac{1}{2}}\left(\|\boldsymbol{v}\|^2 + \delta t\|\nabla\boldsymbol{v}\|^2\right)^{\frac{1}{2}} \\
&= c_1\|\boldsymbol{u}\|_V\|\boldsymbol{v}\|_V,
\end{aligned}$$

where (15) follows by the Cauchy–Schwarz inequality and the definition

$$a_{max} = \max\left(\max_{ij}|a_{ij}|, 1\right)$$

and (16) comes from the fact that

$$\begin{aligned}
\sum_{i,j\leq s}\|\nabla u_i\|\|\nabla v_j\| &= \sum_{i\leq s}\|\nabla u_i\|\sum_{j\leq s}\|\nabla v_j\| \\
&\leq s\|\nabla\boldsymbol{u}\|\|\nabla\boldsymbol{v}\|, && (17)
\end{aligned}$$

where (17) follows from the equivalence ℓ^1 and ℓ^2 norms on $\mathbb{R}^s$.

□

To prove the inf–sup condition we introduce a family of matrices which we will refer to as weakly positive definite. These matrices are defined as follows.

Definition 3.1. *A matrix $A\in\mathbb{R}^{s,s}$ is weakly positive definite given that there exists a $C\in\mathbb{R}^{s,s}$ such that*

$$x^TCx > 0, \quad \forall x\in\mathbb{R}^s, \tag{18}$$

$$x^TCAx > 0, \quad \forall x\in\mathbb{R}^s. \tag{19}$$

Lemma 3.2. *A real square matrix is weakly positive definite if and only if the real eigenvalues are positive.*

The proof of this lemma can be found in [18]. Weakly positive matrices can be seen as a generalization of the diagonally stable matrices discussed in e.g. [10].

In the following calculations we need that there exists a positive number α such that

$$x^T C x \geq \alpha \|x\|^2, \quad \forall x \in \mathbb{R}^s, \tag{20}$$

$$x^T CA x \geq \alpha \|x\|^2, \quad \forall x \in \mathbb{R}^s. \tag{21}$$

This follows since C and CA are positive definite (not necessarily symmetric). To see that (18) leads to (20), let $x = \frac{y}{\|y\|}$ and use linearity to obtain the equivalent statement that $x^T C x > \alpha$ for $\|x\| = 1$. Furthermore, since $x^T C x$ is a continuous function of x and the set defined by $\|x\| = 1$ is compact, $x^T C x$ will have a lower bound and therefore (20) follows from (18). Similarly, (21) follows from (19).

In the next lemma we use that the Runge–Kutta coefficient matrix A is weakly positive definite, i.e., there exists a positive definite matrix C such that CA is positive definite. By Lemma 3.2 this is true as long as the arguments of the eigenvalues of A differ from π. However, for A–stable irreducible Runge–Kutta methods with an invertible A a stronger result is known, namely that the real parts of the eigenvalues of A are positive. To see this notice that the stability function of a Runge–Kutta method is given by

$$R(z) = \frac{det(I - zA + z\mathbf{1}b^T)}{det(I - zA)}.$$

Irreducibility gives that the fraction can not be reduced. Assume that there exists an eigenvalue μ of A with $Re(\mu) < 0$. Then the stability function will have a pole in $z = 1/\mu$, which also lies in the left half plane. This contradicts with the A–stability, which requires that $|R(z)| \leq 1$ for all z in the left half plane. Furthermore, A is invertible, therefore zero is not an eigenvalue. We therefore conclude that the real part of the eigenvalues must be positive.

Lemma 3.3. *There exists a constant c_2 independent of δt such that*

$$\sup_{\boldsymbol{v} \in \boldsymbol{V}} \frac{a(\boldsymbol{u}, \boldsymbol{v})}{\|\boldsymbol{v}\|_{\boldsymbol{V}}} \geq \frac{1}{c_2} \|\boldsymbol{u}\|_{\boldsymbol{V}}, \quad \forall \boldsymbol{u} \in \boldsymbol{V}.$$

Proof. Define

$$\alpha = \min \left(\min_{\|x\|=1} x^T C x, \min_{\|x\|=1} x^T CA x \right).$$

Given $\boldsymbol{u} \in \boldsymbol{V}$, let $\boldsymbol{v} = C^T \boldsymbol{u}$, where C is constructed such that we have (20)-(21) with A being the Runge–Kutta coefficient matrix. Then,

$$\begin{aligned}
\sup_{\boldsymbol{v} \in \boldsymbol{V}} \frac{(\mathcal{A}\boldsymbol{u}, \boldsymbol{v})_{L^2}}{\|\boldsymbol{u}\|_{\boldsymbol{V}} \|\boldsymbol{v}\|_{\boldsymbol{V}}} &\geq \frac{(\mathcal{A}\boldsymbol{u}, C^T \boldsymbol{u})_{L^2}}{\|\boldsymbol{u}\|_{\boldsymbol{V}} \|C^T \boldsymbol{u}\|_{\boldsymbol{V}}} \\
&= \frac{(C\mathcal{A}\boldsymbol{u}, \boldsymbol{u})_{L^2}}{\|\boldsymbol{u}\|_{\boldsymbol{V}} \|C^T \boldsymbol{u}\|_{\boldsymbol{V}}} \\
&= \frac{(C\boldsymbol{u}, \boldsymbol{u})_{L^2} - \delta t((CA \otimes \Delta)\boldsymbol{u}, \boldsymbol{u})_{L^2}}{\|\boldsymbol{u}\|_{\boldsymbol{V}} \|C^T \boldsymbol{u}\|_{\boldsymbol{V}}} \\
&\geq \frac{1}{\|C\|} \frac{(C\boldsymbol{u}, \boldsymbol{u})_{L^2} + \delta t(CA\nabla \boldsymbol{u}, \nabla \boldsymbol{u})_{L^2}}{\|\boldsymbol{u}\|_{\boldsymbol{V}} \|\boldsymbol{u}\|_{\boldsymbol{V}}} \\
&\geq \frac{\alpha}{\|C\|} \frac{\|\boldsymbol{u}\|_{L^2}^2 + \delta t \|\nabla \boldsymbol{u}\|_{L^2}^2}{\|\boldsymbol{u}\|_{\boldsymbol{V}}^2} \\
&= \frac{\alpha}{\|C\|} \\
&> 0,
\end{aligned}$$

where $\|C\|$ is the L^2 matrix norm. Hence, $c_2 = \frac{\|C\|}{\alpha}$ in (13).

□

Finally, (14) follows by an argument similar to the proof of (13) in Lemma (3.3). First, we notice that A^T and A share the same set of eigenvalues, and therefore A^T is also weakly positive definite. By Lemma 3.2 we now have that there exists a D such that both D and DA^T are positive definite. Further let $\boldsymbol{v}$ be given and let $\boldsymbol{u} = D^T\boldsymbol{v}$. Then we have

$$\begin{aligned} a(\boldsymbol{u},\boldsymbol{v}) &= a(D^T\boldsymbol{v},\boldsymbol{v}) = ((I - A\otimes\Delta)D^T\boldsymbol{v},\boldsymbol{v}) = (D^T - AD^T\otimes\Delta)\boldsymbol{v},\boldsymbol{v}) \\ &= (\boldsymbol{v},D\boldsymbol{v}) + (\nabla\boldsymbol{v}, DA^T\nabla\boldsymbol{v}) > 0. \end{aligned}$$

We therefore conclude that $\mathcal{A}$ is an isomorphism (11).

4. The Preconditioner

The preconditioner for the continuous operator is constructed based on the weighted Sobolev spaces. The discrete preconditioner can be viewed as an operator acting on discrete subspaces. Other works taking the same approach are [1], [9], and [17].

Let $\mathcal{B} : \boldsymbol{V}^* \to \boldsymbol{V}$ be

$$\mathcal{B} = (I - \delta t\,\mathrm{diag}(A)\otimes\Delta)^{-1}, \tag{22}$$

where we assume that the diagonal of A is always strictly positive. It follows from the definition of $\boldsymbol{V}$ and $\boldsymbol{V}^*$ that

$$\|\mathcal{B}\|_{\mathcal{L}(\boldsymbol{V}^*,\boldsymbol{V})} \le d_1 \quad \text{and} \quad \|\mathcal{B}^{-1}\|_{\mathcal{L}(\boldsymbol{V},\boldsymbol{V}^*)} \le d_2, \tag{23}$$

where d_1 and d_2 are independent of δt. Hence,

$$\mathcal{B}\mathcal{A} : \boldsymbol{V} \to \boldsymbol{V}$$

and

$$\|\mathcal{B}\mathcal{A}\|_{\mathcal{L}(\boldsymbol{V},\boldsymbol{V})} \le c_1 d_1 \quad \text{and} \quad \|(\mathcal{B}\mathcal{A})^{-1}\|_{\mathcal{L}(\boldsymbol{V},\boldsymbol{V})} \le c_2 d_2. \tag{24}$$

This means that the condition number of the continuous operator is bounded, i.e.,

$$\kappa(\mathcal{B}\mathcal{A}) = \|\mathcal{B}\mathcal{A}\|_{\mathcal{L}(\boldsymbol{V},\boldsymbol{V})}\|(\mathcal{B}\mathcal{A})^{-1}\|_{\mathcal{L}(\boldsymbol{V},\boldsymbol{V})} \le c_1 d_1 c_2 d_2. \tag{25}$$

In the following we study the condition number in the discrete case. Let Ω_h be a triangulation of Ω. Furthermore, let V_h be a finite element space $V_h \subset H_0^1$, while $\boldsymbol{V}_h = (V_h)^s \subset \boldsymbol{H}_0^1$ is the corresponding vector finite element space. The discrete counterpart of $\mathcal{A}$ and $\mathcal{B}$ are defined by

$$(\mathcal{A}_h\boldsymbol{u}_h,\boldsymbol{v}_h) = (\boldsymbol{u}_h,\boldsymbol{v}_h) + \delta t(A\nabla\boldsymbol{u}_h,\nabla\boldsymbol{v}_h), \qquad \forall \boldsymbol{u}_h,\boldsymbol{v}_h \in \boldsymbol{V}_h, \tag{26}$$

$$(\mathcal{B}_h^{-1}\boldsymbol{u}_h,\boldsymbol{v}_h) = (\boldsymbol{u}_h,\boldsymbol{v}_h) + \delta t\,\mathrm{diag}(A)(\nabla\boldsymbol{u}_h,\nabla\boldsymbol{v}_h), \qquad \forall \boldsymbol{u}_h,\boldsymbol{v}_h \in \boldsymbol{V}_h. \tag{27}$$

Since $\boldsymbol{V}_h$ is a subspace of $\boldsymbol{V}$ we get that

$$\kappa(\mathcal{B}_h\mathcal{A}_h) \le \kappa(\mathcal{B}\mathcal{A}). \tag{28}$$

Let $\tilde{B}_i$ be a cheap approximation of $B_i = (I - \delta t a_{ii}\Delta_h)^{-1}$ constructed by, e.g., multigrid or domain decomposition. This is the standard preconditioner which is used for the implicit Euler step. It is then well known that $\tilde{B}_i$ may be constructed such that it is symmetric and spectrally equivalent to B_i, i.e.,

$$c_3(\tilde{B}_i^{-1}v,v) \le ((I - \delta t a_{ii}\Delta_h)v,v) \le c_4(\tilde{B}_i^{-1}v,v), \quad \forall v \in V_h. \tag{29}$$

Let c_3 and c_4 be chosen such that (29) is valid for all i. Let further the full preconditioner be denoted

$$\tilde{\mathcal{B}}_h = \begin{pmatrix} \tilde{B}_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \tilde{B}_s \end{pmatrix}.$$

We have that the condition number fulfills the Cauchy–Schwartz like property $\kappa(CD) \leq \kappa(C)\kappa(D)$ for two matrices C and D, since

$$\kappa(CD) = \|CD\|\|(CD)^{-1}\| \leq \|C\|\|D\|\|C^{-1}\|\|D^{-1}\| = \kappa(C)\kappa(D).$$

Therefore we get

$$\kappa(\tilde{\mathcal{B}}_h\mathcal{A}_h) \leq \kappa(\tilde{\mathcal{B}}_h\mathcal{B}_h^{-1})\kappa(\mathcal{B}_h\mathcal{A}_h)$$

By using (25) and (29), we get,

$$\kappa(\tilde{B}_h\mathcal{A}_h) \leq \frac{c_1 d_1 c_2 d_2 c_4}{c_3}. \tag{30}$$

5. Numerical Experiments

In this section we test our preconditioner in a series of numerical experiments. We test three families of fully implicit Runge–Kutta schemes, Gauss, Radau, and Lobatto. The different families have different properties when it comes to the order of the approximation and the stability. We will only give a brief description here, and refer to [8] for a more thorough description.

By using Gauss quadrature rules, we get the implicit Runge–Kutta Gauss methods. They have a global error estimate of $\mathcal{O}(\delta t^{2s})$, where s is the number of quadrature points. Notice that the single–node Gauss method is the implicit midpoint method and it is equal to the Crank–Nicolson scheme in our example.

Using Radau quadrature, which requires one end-point among the quadrature nodes, the global error estimate is $\mathcal{O}(\delta t^{2s-1})$. These methods will be named RadauIA and RadauIIA, where the start or the endpoint is among the quadrature nodes, respectively. Notice that implicit Euler is the single–node method where the endpoint is the quadrature node.

Finally, one can require that both the start and end points are quadrature nodes, which is called Lobatto quadrature. There exist three different sub families of the Lobatto scheme, where the coefficient matrix A differs among them. They have different properties regarding, e.g., stability, but they all have a global error estimate of $\mathcal{O}(\delta t^{2s-2})$. Two of the sub families have one explicit step. These methods are not studied here since they have a singular A-matrix. We will only consider the so called LobattoIIIC methods.

All the methods are A–stable, but only the Radau methods and the LobattoIIIC methods are L–stable, which is an attractive property for stiff problems, like a semidiscretized heat equation. In addition, the RadauIIA and the LobattoIIIC methods are stiffly accurate.

It is well-known from standard ODE theory that in the case of very stiff problems, where $\delta t \to 0$, while $z = \lambda\delta t \to \infty$ where λ is an eigenvalue of $I_h^{-1}\Delta_h$, the order of the implicit Runge–Kutta schemes is reduced. This is known as B-convergence, and is more thoroughly described in [8]. This does however only affect the order of convergence for the Runge–Kutta scheme, and not the performance of our preconditioner. In our numerical tests, we did not experience order reduction.

5.1. **The condition number.** First we want to verify numerically the order–optimality of the preconditioner with respect to the spatial discretization parameter h and the timestep δt. This is done for a three node RadauIIA method. We solve a 1D problem, where linear finite elements are used in space. The preconditioner is the inverse of the submatrices on the diagonal. The results are presented in Table 1 and Figure 1.

$\delta t/h$	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}
0.1	8.23	13.40	14.93	15.32	15.42	15.44	15.45	15.45
0.05	5.49	11.79	14.44	15.19	15.38	15.43	15.44	15.45
0.02	2.90	8.55	13.14	14.82	15.29	15.41	15.44	15.45
0.01	1.91	5.79	11.37	14.23	15.13	15.37	15.43	15.44
0.005	1.43	3.58	8.89	13.18	14.82	15.29	15.41	15.44
0.002	1.16	1.99	5.28	10.72	13.96	15.05	15.35	15.42
0.001	1.08	1.47	3.24	8.08	12.71	14.67	15.25	15.40

TABLE 1. The condition number $\kappa(\mathcal{BA})$ for the 1D problem (1)–(3) using linear finite elements in space, and three node RadauIIA method in time. The preconditioner is constructed by inverting the diagonal blocks exactly.

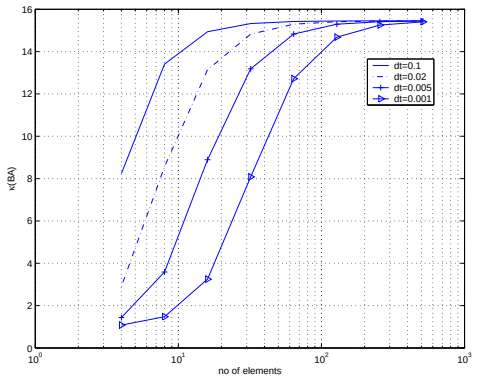

FIGURE 1. The condition number $\kappa(\mathcal{BA})$ for the 1D problem (1)–(3) using linear finite elements in space, and three node RadauIIA method in time. The preconditioner is inverted exactly.

If the timestep δt is sufficiently small compared to the spatial discretization parameter h, the Helmholtz operator is close to the mass matrix, and the condition number is small. For sufficiently small spatial discretization compared to the timestep, the condition number $\kappa(\mathcal{BA})$ of the preconditioned system seems to reach an asymptotic value of about 15.4. The next section concerns the asymptotic value for this and other schemes in both 1D and 2D using multigrid preconditioners. In [19] we have also shown experiments in 3D.

The results clearly confirm that the preconditioner is order–optimal with respect to both the spatial discretization parameter h and the timestep δt.

5.2. **Multigrid preconditioning.** We will use multigrid to approximate the preconditioner. All computations will be done on a domain $\Omega = (0,1)^d$, where d is the number of spatial dimensions. A sequence of meshes is constructed by uniform

Method	Exact 1D	MG 1D	MG 2D
Gauss 1 node (Implicit midpoint)	1.0	1.9	1.1
Gauss 2 nodes	4.8	9.1	5.2
Gauss 3 nodes	11.8	22.0	12.8
Gauss 4 nodes	22.4	41.2	24.2
Gauss 5 nodes	37.2	67.8	40.0
Gauss 6 nodes	56.6	102.0	60.4
RadauIA 1 node	1.0	1.9	1.1
RadauIA 2 nodes	1.5	2.6	1.5
RadauIA 3 nodes	7.2	13.6	7.8
RadauIIA 1 node (Implicit Euler)	1.0	1.9	1.1
RadauIIA 2 nodes	6.8	12.9	7.4
RadauIIA 3 nodes	15.4	29.0	16.8
RadauIIA 4 nodes	27.1	50.1	29.3
RadauIIA 5 nodes	41.2	75.2	44.3
RadauIIA 6 nodes	57.5	104	61.5
LobattoIIIC 2 nodes	1.3	1.9	1.4
LobattoIIIC 3 nodes	11.2	21.4	12.2
LobattoIIIC 4 nodes	21.6	40.6	23.5

TABLE 2. Estimates of the condition number $\kappa(\mathcal{BA})$ for various implicit Runge–Kutta schemes.

refinement of a 2 or 2×2 partition of the domain Ω. The preconditioner $\tilde{\mathcal{B}}$ is computed using a standard V-cycle with a symmetric Gauss-Seidel smoother. Gaussian elimination is used as the coarse grid solver.

We estimate the condition number using one multigrid V–cycle as the preconditioner. The results are shown in Table 2 for both 1D and 2D. The discretization parameters δt and h are chosen such that the condition number is close to the asymptotic value. In our case $\delta t = 0.1$ and $h = 2^{-9}$ have been appropriate. The results for multigrid in 2D are better than the results for multigrid in 1D, and this seems a bit strange. We have tested this multigrid method also in the case of a Poisson equation and similar results were obtained here. However, it is probably possible to tune the parameters in the 1D multgrid method such that it performs better, but we have not done this, since the results anyway are good.

The single stage methods are included as a point of reference. In 2D, the multigrid preconditioner is much better than in 1D, and the condition numbers are roughly 10% higher than the exact 1D case.

Clearly our preconditioner is not optimal with respect to the number of quadrature nodes in the Runge–Kutta scheme. In fact, it seems that the growth in the condition number with respect to s is slightly above quadratic. But remember that an increase in the number of nodes by one, implies an increase in the order of two. So an increase in the number of iterations, may be subordinate to the decrease in the number of required timesteps. This is what we will investigate in the next example. Notice also that we in [19] studied Gauss-Seidel preconditioners which have far better performance with respect to s.

5.3. **Iteration count for a test problem.** Finally, we compare the actual CPU time for a given test problem. We solve (1)–(3), with a source term f such that the exact solution is

$$u(x, y, t) = \sin(\omega_x x) \sin(\omega_y y) \sin(\omega_t \pi t) e^{xy}, \quad (\omega_x, \omega_y, \omega_t) = (\pi, \pi, 20.5\pi).$$

		With MG prec			No prec		
Method	δt	wct	iter	lst	wct	iter	lst
Gauss 1 node	1/5000	87.4	2.9	22.7	92.8	18.4	27.50
Gauss 2 nodes	1/142	11.0	14.7	6.4	169	1184	163.6
Gauss 3 nodes	1/47	8.5	24.4	5.7	280	3565	623
Gauss 4 nodes	1/26	8.8	34.8	6.3	—	>5000	—
RadauIIA 1 node	5.0e-7	$3e4$	3	9085	$2e4$	2	1607
RadauIIA 2 nodes	1/440	34.5	15	20.2	277	607	262
RadauIIA 3 nodes	1/68	14.0	30	10.0	488	3684	817
RadauIIA 4 nodes	1/30	12.5	46.8	9.8	—	>5000	—
LobattoIIIC 2 nodes	1/5000	387.5	14.7	227.5	530	105	515
LobattoIIIC 3 nodes	1/140	33.5	37.1	25.5	953	2902	938
LobattoIIIC 4 nodes	1/43	23.4	64.5	19.5	—	>5000	—

TABLE 3. The total computational time measured in minutes (wct), the time spent to solve the linear system (lst), and the average number of iterations (for all time steps) for solving the heat equation (1)–(3) for schemes with various number of stages. The discretization parameters are chosen such that the error from the discretizations are approximately 10^{-5}. The preconditioner is approximated using one multigrid V–cycle. For comparison, the same simulations are also done without precondtioner.

The high number of oscillations in time is to generate a certain degree of complexity in time. We solve the problem from $T = 0.0$ to $T = 1.0$ on the unit square. In space we discretize using linear finite elements. Both the element size h and the time–step δt is chosen such that the error is of order 10^{-5}, measured in the following discrete $L^2(0, T; L^2)$ norm,

$$\|e\|_{L^2(0,T,L^2)} = (\sum_{k=0}^{n} e(t_k)^T M e(t_k)\, \delta t)^{1/2},$$

where $e(t_k)$ is the error vector at nodal points at time t_k, M is the mass matrix and n is the number of time steps. In space we discretized using 2^{16} bilinear finite elements over the entire domain.

The preconditioner is one geometric multigrid V–cycle, and the linear system is solved using GMRES with restart after 5 search vectors (for RadauIIA and Gauss with 1 node we used conjugated gradients) with the stopping criterion that $\|r_k\| < 10^{-7}$. The result is displayed in Table 3. Notice that the residual is only evaluated before the restart. This means that the system is possibly over–iterated, but the computational time is in general smaller due to the high cost of evaluating the residual every iteration.

The results are computed on a Linux machine with an AMD Athlon 64 2.2GHz processor with 2GB RAM using the C++ library Diffpack.

The number of iterations stated is an average over all the iterations. The three-node Gauss scheme is the fastest, while the four-node Gauss is almost equally fast. In general, the higer-order schemes is more efficient the lower-order schemes for all the three classes of Runge-Kutta schemes.

The cost of doing a preconditioned GMRES iteration is approximately 2.5 the cost of doing an unpreconditioned iteration. The cost of evaluating the right hand side for each timestep is equal to several preconditioned iterations due to the complexity of the function f. This is noticable for the schemes where the number of

required iterations is low, where the time to solve the linear systems (lst) is much smaller then the total solution time (wct).

Note that this is only an indication that high order Runge–Kutta methods may be efficient compared to low order methods. But which solver is the most efficient depends on several properties like the regularity of the solution, the required accuracy, the implementation, and other properties.

6. Final remarks

In this paper we have shown that the systems arising from fully implicit Runge–Kutta schemes applied to parabolic equations can be efficiently preconditioned with a block preconditioner. The block preconditioner is block diagonal and has blocks that are standard elliptic preconditioners. Such preconditioner are well known to be order–optimal when constructed by, e.g., multigrid or domain decomposition methods. The proposed preconditioner is proven to be order–optimal when constructed from order–optimal preconditioner for backward Euler scheme.

In several numerical experiments we have demonstrated that the condition number for the preconditioned systems are bounded. We have also seen that higher–order methods are beneficial, when using efficient preconditioners, even for problems with relatively fast dynamics and modest accuracy requirements.

Acknowledgment. The authors are grateful to Professors Brynjulf Owren and Syvert P. Nørsett at Departement of Mathematical Science, Norwegian University of Science and Tehcnology, Norway, for many useful discussions.

References

[1] D.N. Arnold, R.S. Falk, and R. Winther. Preconditioning discrete approximations of the Reissner–Mindlin plate model. *Math. Modelling Numer. Anal.*, 31:517–557, 1997.

[2] U. M. Asher and L. R. Petzold. *Computer Methods for Ordinary Differential Equations*. SIAM, Philadelphia, 1998.

[3] I. Babuska and A. K. Aziz. Survey lectures on the mathematical foundations on the finite element method. In A. K. Aziz, editor, *The Mathematical Foundation of the Finite Element Method with Applications to Partial Differential Equations*, pages 3–363. Academic Press, New York - London, 1972.

[4] J. Bergh and J. Löfström. *Interpolation spaces*. Springer Verlag, 1976.

[5] J. H. Bramble and P. H. Sammon. Efficient higher order single step methods for parabolic problems: Part i. *Math. Comp.*, 35:655–677, 1980.

[6] Lawrence C. Evans. *Partial Differential Equations*. Number 19. American Mathematical Society, 1998.

[7] B. Gustafsson and W. Kress. Deferred correction methods for initial value problems. *BIT*, 41:986–995, 2001.

[8] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II - Stiff and Differential-Algebraic Problems*. Springer Verlag, 2nd edition, 1996.

[9] E. Haug and R. Winther. A domain embedding preconditioner for the Lagrange multiplier system. *Math. Comp*, 69:65–82, 1999.

[10] H. Hu. A projective method for rescaling a diagonally stable matrix to be positive definite. *SIAM J. Matrix Anal. Appl.*, 13(4):1255–1263, 1992.

[11] F. Iavernaro and F. Mazzia. Solving ordinary differential equations by generalized adams methods: properties and implementation techniques. *Appl. Num. Math.*, 28 (2-4):107–126, 1998.

[12] L.O. Jay. Inexact simplified newton iterations for implicit runge-kutta methods. *SIAM J. Numer. Anal.*, 38:1369–1388, 2000.

[13] L.O. Jay and T. Braconnier. A parallelizable preconditioner for the iterative solution of implicit runge-kutta type methods. *J. Comput. Appl. Math.*, 111:63–76, 1999.

[14] J. Van lent and S. Vandewalle. Multigrid methods for Implicit Runge-Kutta and Boundary Value Method Discretizations of PDEs. *SIAM J. Sci. Comput*, 27:67–92, 2005.

[15] K.-A. Mardal and T. K. Nilssen. Reuse of standard preconditioners for higher–order time discretizations of parabolic pdes. *Journal of Numerical Mathematics*, 14(2):103–122, 2006.

[16] K.-A. Mardal, J. Sundnes, H. P. Langtangen, and A. Tveito. *Advanced Topics in Computational Partial Differential Equations - Numerical Methods and Diffpack Programming (Langtangen, H.P. and Tveito, A. (eds))*, chapter Block preconditioning and Systems of PDEs, pages 199–236. Lecture Notes in Computational Science and Engineering. Springer-Verlag, 2003.

[17] K.-A. Mardal and R. Winther. Uniform preconditioners for the time dependent Stokes problem. *Numer. Math.*, 98(2):305–327, 2004.

[18] T. K. Nilssen. Weakly positive definite matrices. Simula Research Laboratory, Research Report 07–2005.
URL: http://www.simula.no/departments/scientific/publications/Nilssen.2005.1.

[19] G. A. Staff, Kent-A. Mardal, and T. K. Nilssen. Preconditioning of fully implicit Runge-Kutta schemes for parabolic PDEs. *Modeling, Identification and Control*, 27(2):109–123, 2006.

[20] V. Thomée. *Galerkin Finite Element Methods for Parabolic Problems*. Springer-Verlag, 2nd edition, 1997.