

A NEW BLOCK PRECONDITIONER FOR IMPLICIT RUNGE–KUTTA METHODS FOR PARABOLIC PDE PROBLEMS

MD MASUD RANA*, VICTORIA E. HOWLE†, KATHARINE LONG‡, ASHLEY MEEK§,
AND WILLIAM MILESTONE¶

Abstract. A new preconditioner based on a block LDU factorization with algebraic multigrid subsolves for scalability is introduced for the large, structured systems appearing in implicit Runge–Kutta time integration of parabolic partial differential equations. This preconditioner is compared in condition number and eigenvalue distribution, and in numerical experiments with others in the literature: block Jacobi, block Gauss–Seidel, and the optimized block Gauss–Seidel method of Staff, Mardal, and Nilssen [*Modeling, Identification and Control*, 27 (2006), pp. 109–123]. Experiments are run on two test problems, a 2D heat equation and a model advection-diffusion problem, using implicit Runge–Kutta methods with two to seven stages. We find that the new preconditioner outperforms the others, with the improvement becoming more pronounced as spatial discretization is refined and as temporal order is increased.

1. Introduction. Explicit time integrators for parabolic PDEs are subject to the restrictive timestep limit $h_t \lesssim h_x^2$, so A-stable integrators are essential. It is well known that although there are no A-stable explicit linear multistep methods and implicit multistep methods cannot be A-stable beyond order two, there exist A-stable and L-stable implicit Runge–Kutta (IRK) methods at all orders [2, 9]. IRK methods offer an appealing combination of stability and high order; however, these methods are not widely used for PDEs because they lead to large, strongly coupled linear systems. An s -stage IRK system has s times as many degrees of freedom as the systems resulting from backward Euler or implicit trapezoidal rule discretization applied to the same equation set. Order-optimal preconditioners for such systems have been investigated in a series of papers [5, 7]. In this paper we introduce a new block preconditioner for IRK methods, based on an LDU factorization. Solves on individual blocks are accomplished using a multigrid algorithm. We demonstrate the effectiveness of this preconditioner on two test problems. The first is a simple heat equation, and the second is a model advection-diffusion problem known as the double-glazing problem. We find that our preconditioner is scalable (h_x -independent) and yields better timing results than other preconditioners currently in the literature.

2. Implicit Runge–Kutta Methods. IRK methods are described in detail in, for example, [9]. The general s -stage IRK method for $u' = f(t, u)$ requires the solution of s simultaneous equations

$$K_i = f \left(t_n + c_i h_t, u_n + h_t \sum_{j=1}^s a_{ij} K_j \right) \quad \text{for } i = 1 \text{ to } s$$

*Corresponding author (co-authors listed alphabetically); Texas Tech University (md-masud.rana@ttu.edu)

†Texas Tech University (victoria.howle@ttu.edu)

‡Texas Tech University (katharine.long@ttu.edu)

§Adams State University (ameek@adams.edu)

¶Texas Tech University (William.Milestone@ttu.edu)

for the stage variables K_i , $i = 1$ to s . The stage variables are then used to update the solution u ,

$$u_{n+1} = u_n + h_t \sum_{i=1}^s b_i K_i.$$

The coefficients that define a given method are summarized in a Butcher table:

$$\begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array} = \frac{c}{b^T} \bigg| \frac{A}{b^T}.$$

We list some common IRK methods with their order and stability properties in Table 2.1. As our goal is to use higher order L-stable methods, throughout this paper we consider Radau IIA and Lobatto IIIC methods. Of the common L-stable methods, these provide the highest order for a given number of stages, that is, for a given cost.

Table 2.1: Common IRK Methods

IRK Methods	Order	Stability
Gauss–Legendre (s)	$2s$	A-stable
Radau IA (s)	$2s - 1$	L-stable
Radau IIA (s)	$2s - 1$	L-stable
Lobatto IIIA (s)	$2s - 2$	A-stable
Lobatto IIIC (s)	$2s - 2$	L-stable
Miller DIRK (2)	2	L-stable
Miller DIRK (3)	2	L-stable
Crouzeix SDIRK(2)	2	L-stable
Crouzeix SDIRK(3)	3	L-stable

3. Formulation of the Test Problems. We consider two test problems. The first is a simple heat equation, and the second is a double-glazing advection-diffusion problem.

3.1. Heat Equation. As a first simple test problem, we consider the heat equation on some spatial domain Ω over the time interval $[0, T]$ with Dirichlet boundary conditions:

$$\begin{aligned} (3.1) \quad & u_t = \nabla^2 u \quad \text{in } \Omega \times [0, T] \\ & u = 0 \quad \text{on } \partial\Omega \\ & u(\bar{x}, 0) = u_0(\bar{x}). \end{aligned}$$

We first discretize in time using an IRK method. We write u_n for $u(\bar{x}, nh)$ and introduce stage variables $K_i = K_i(\bar{x})$ for $i = 1$ to s . In strong form, the stage equations for an s -stage IRK method are:

$$\begin{aligned} K_i &= \nabla^2 u_n + h_t \sum_{j=1}^s a_{ij} \nabla^2 K_j \quad \text{in } \Omega \\ K_i &= 0 \quad \text{on } \partial\Omega \quad \text{for } i = 1 \text{ to } s. \end{aligned}$$

We discretize this system of time-independent PDEs using the finite element method. Adopting test functions $\hat{K}_i \in H_0^1$, we obtain a weak form as usual by multiplying the i -th stage equation by the test function \hat{K}_i , integrating, and applying Green's identity. This leads to the equations:

$$\begin{aligned} \int_{\Omega} \left[\hat{K}_i K_i + \nabla \hat{K}_i \cdot \nabla u_n + h_t \sum_{j=1}^s a_{ij} \nabla \hat{K}_i \cdot \nabla K_j \right] d\Omega - \int_{\partial\Omega} \hat{K}_i \hat{\mathbf{n}} \cdot \nabla u_n d\Gamma \\ - h_t \sum_{j=1}^s a_{ij} \int_{\partial\Omega} \hat{K}_i \hat{\mathbf{n}} \cdot \nabla K_j d\Gamma = 0, \end{aligned}$$

which must hold for all $\hat{K}_i \in H_0^1$. Note that the boundary terms vanish due to the boundary restriction of \hat{K}_i to H_0^1 . Therefore we need to find $K_i \in H_0^1$ such that

$$\int_{\Omega} \left[\hat{K}_i K_i + \nabla \hat{K}_i \cdot \nabla u_n + h_t \sum_{j=1}^s a_{ij} \nabla \hat{K}_i \cdot \nabla K_j \right] d\Omega = 0 \quad \forall \hat{K}_i \in H_0^1.$$

Converting to weak form and discretizing with finite element basis functions ϕ_j , we get the following linear system, which must be solved for the stage variables K_i :

$$(3.2) \quad \begin{bmatrix} M + a_{11}h_tF & a_{12}h_tF & \cdots & a_{1s}h_tF \\ a_{21}h_tF & M + a_{22}h_tF & \cdots & a_{2s}h_tF \\ \vdots & \vdots & \ddots & \vdots \\ a_{s1}h_tF & a_{s2}h_tF & \cdots & M + a_{ss}h_tF \end{bmatrix} \begin{bmatrix} K_1 \\ K_2 \\ \vdots \\ K_s \end{bmatrix} = - \begin{bmatrix} Fu_n \\ Fu_n \\ \vdots \\ Fu_n \end{bmatrix},$$

where the elements of the matrices M and F are given by

$$\begin{aligned} M_{kl} &= \int \phi_k \phi_l = O(h_x^d) \\ F_{kl} &= \int \nabla \phi_k \cdot \nabla \phi_l = O(h_x^{d-2}), \end{aligned}$$

and where d is the spatial dimension of the problem. Thus we have an $sN \times sN$ linear system to solve at each time step; this is the system that we need to precondition.

3.2. Double-Glazing Advection-Diffusion Problem. As a second test problem, we consider a model advection-diffusion problem known as the double-glazing problem. A time-independent version of this problem is described in detail in [1]; we modify it here to a time-dependent problem. This is a simple model for advective-diffusive transport in a cavity $\Omega = [-1, 1] \times [-1, 1]$, where one wall is hot. The equation and boundary conditions are

$$(3.3) \quad \begin{aligned} u_t + (\bar{w} \cdot \nabla) u - \epsilon \nabla^2 u &= 0 \quad \text{in } \Omega \times [0, T] \\ u &= 0 \quad \text{on } \partial\Omega_N \cup \partial\Omega_W \cup \partial\Omega_S \\ u &= 1 \quad \text{on } \partial\Omega_E \\ u(\bar{x}, 0) &= u_0(\bar{x}), \end{aligned}$$

where $\epsilon > 0$ and $\partial\Omega_N$, $\partial\Omega_W$, $\partial\Omega_S$ and $\partial\Omega_E$ are the North, West, South, and East walls of Ω . For initial conditions we take $u_0(\bar{x}) = 0$ on $\Omega \setminus \partial\Omega_E$; the solution evolves

towards the steady solution described in [1]. The transport is advection dominated when $\epsilon \ll 1$ and diffusion dominated when $\epsilon \gtrsim 1$. For the wind $\bar{w}(\bar{x})$, we use the circulating interior flow described in [1]. As before, we discretize in time using an IRK method then convert the resulting time-independent PDEs to weak form and discretize in space using stabilized finite elements. This results in a linear system for the stage variables K_i of the same structural form as (3.2).

4. Preconditioning IRK methods applied to parabolic PDEs. In this paper, we develop a block upper triangular preconditioner and a block lower triangular preconditioner for systems of the form (3.2).

Let A be the s -stage IRK coefficient matrix, that is, the matrix comprising the elements in the upper-right block of the Butcher table for the given IRK method.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1s} \\ a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ a_{s1} & a_{s2} & \cdots & a_{ss} \end{bmatrix}.$$

Given an $m \times n$ matrix B and a matrix C , recall that their Kronecker product $B \otimes C$ is given by

$$B \otimes C = \begin{bmatrix} b_{11}C & b_{12}C & \cdots & b_{1n}C \\ b_{21}C & b_{22}C & \cdots & b_{2n}C \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1}C & b_{m2}C & \cdots & b_{mn}C \end{bmatrix}.$$

Using this notation, we can write the block matrix (3.2) arising from the FEM discretization as

$$\mathcal{A} = I_s \otimes M + h_t A \otimes F,$$

where I_s is the $s \times s$ identity matrix.

In this paper, we consider several block preconditioners \mathcal{P} of the general form

$$(4.1) \quad \mathcal{P} = I_s \otimes M + h_t \tilde{A} \otimes F,$$

where \tilde{A} is either a diagonal or triangular matrix referred as the *preconditioner coefficient matrix*. We can, for example, define a block Jacobi preconditioner \mathcal{P}_J and a block Gauss–Seidel preconditioner \mathcal{P}_{GSL} using this notation as follows:

$$(4.2) \quad \begin{aligned} \mathcal{P}_J &= I_s \otimes M + h_t \tilde{A}_J \otimes F \\ \mathcal{P}_{GSL} &= I_s \otimes M + h_t \tilde{A}_{GSL} \otimes F, \end{aligned}$$

where \tilde{A}_J is the diagonal part of A and \tilde{A}_{GSL} is the lower triangular part of A . These preconditioners and their order optimality have been extensively studied in Mardal et al. [5] and in G.A. Staff et al. [7]. We now introduce two new *LDU*-based block triangular preconditioners for \mathcal{A} .

4.1. *LDU*-based block triangular preconditioners. It has been shown in [7] that if all blocks of \mathcal{A} are well preconditioned by a preconditioner \mathcal{P} of the form (4.1),

then all of the eigenvalues of the preconditioned system will be clustered, and the condition number of the preconditioned system can be approximated by $\kappa(\mathcal{P}^{-1}\mathcal{A}) \approx \kappa(\tilde{A}^{-1}A)$, where \tilde{A} is the preconditioner coefficient matrix. In other words, if we make \tilde{A} an effective preconditioner for the Butcher coefficient matrix A , then \mathcal{P} can be expected to be an effective (left) preconditioner for the system \mathcal{A} . (In practice, we use the same motivation in developing left and right preconditioners.) We begin, therefore, by preconditioning A .

For the development of our preconditioners, we assume that the IRK coefficient matrix A is invertible and has nonsingular leading principal submatrices. It is proven in [2] that the Butcher coefficient matrices for Gauss–Legendre, Radau IA, Radau IIA, and Lobatto IIIC are nonsingular for all s . We have confirmed that the leading principal submatrices are all nonsingular for the coefficient matrices arising from Gauss–Legendre, Lobatto IIIC, Radau IA, and Radau IIA IRK methods for all stages that we use in this paper: $s = 2$ through $s = 7$. Given these conditions, A can be factored into $A = LDU$ without pivoting. We compute formally the LDU factorization $A = LDU$ without pivoting

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1s} \\ a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ a_{s1} & a_{s2} & \cdots & a_{ss} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{s1} & l_{s2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} d_{11} & & & \\ & d_{22} & & \\ & & \ddots & \\ & & & d_{ss} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & \cdots & u_{1s} \\ & 1 & \cdots & u_{2s} \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix}$$

and consider $\tilde{A}_{DU} = DU$ and $\tilde{A}_{LD} = LD$ as preconditioners for A . Since $A = LDU$, we have

$$\begin{aligned} A &= LDU \\ A(DU)^{-1} &= L \\ \text{and} \\ (LD)^{-1}A &= U. \end{aligned}$$

The eigenvalues of U and L are all one, thus we expect \tilde{A}_{DU} to be a good right preconditioner and \tilde{A}_{LD} to be a good left preconditioner for A . Since the eigenvalues (though not the condition numbers) of the product of two matrices do not depend on the order of the product, we consider both matrices as left and right preconditioners in our numerical experiments.

Using the condition number motivation from [7], we now define block upper and lower triangular preconditioners for \mathcal{A} of the form:

$$\begin{aligned} \mathcal{P}_{DU} &= I_s \otimes M + h_t \tilde{A}_{DU} \otimes F \\ \mathcal{P}_{LD} &= I_s \otimes M + h_t \tilde{A}_{LD} \otimes F. \end{aligned} \tag{4.3}$$

4.2. Application of the Preconditioners. Applying the block Jacobi preconditioner \mathcal{P}_J to a vector v is straightforward, and for an s -stage IRK scheme involves s subsolves. The subsolve for the j th block has the form

$$(M + h_t \tilde{A}_{jj} F) w_j = v_j.$$

In practice and for efficiency, we use a single algebraic multigrid (AMG) V-cycle for each subsolve for the block w_j .

Application of the block upper triangular preconditioner \mathcal{P}_{DU} is done via back substitution and again involves s subsolves. The subsolve on the j th block in this case is of the form

$$\left(M + h_t \tilde{A}_{jj} F\right) w_j = \left[v_j - \sum_{k=j+1}^s h_t \tilde{A}_{jk} F w_k \right].$$

In practice, we again use a single AMG V-cycle for each subsolve.

Application of the block lower triangular preconditioners \mathcal{P}_{GSL} and \mathcal{P}_{LD} is done via forward substitution and again involves s subsolves. The subsolves on the j th block in this case are of the form

$$\left(M + h_t \tilde{A}_{jj} F\right) w_j = \left[v_j - \sum_{k=1}^{j-1} h_t \tilde{A}_{jk} F w_k \right].$$

As with the previous preconditioners, we use a single AMG V-cycle for each subsolve.

4.3. Analysis of the Preconditioners. The preconditioners have been chosen because of their intended effect on the condition number of the preconditioned system. Although, as expected, the condition number is not always a good indicator of performance for GMRES. In Table 4.1, we examine the 2-norm condition number for the unpreconditioned system \mathcal{A} and for \mathcal{A} preconditioned on the right by \mathcal{P}_J , \mathcal{P}_{GSL} , \mathcal{P}_{DU} , and \mathcal{P}_{LD} . The system \mathcal{A} is from a 2D heat equation using an s -stage Radau IIA IRK method for s ranging from 2 to 7. In all of these results, we have chosen $h_x = 2^{-3}$ and $h_t = h_x^{\frac{p+1}{2s-1}}$, where $p = 2$ is the degree of the Lagrange polynomial basis functions in space. The preconditioners are constructed exactly. All of the preconditioners significantly reduce the condition number, with \mathcal{P}_{GSL} and \mathcal{P}_{LD} giving the most improvement. \mathcal{P}_{GSL} gives the greatest improvement on the condition number for the lower-order stages $s = 2$ and $s = 3$, and \mathcal{P}_{LD} gives the greatest improvement for the higher-order stages $s = 3$ through $s = 7$.

Table 4.1: Condition numbers of right-preconditioned matrices with preconditioners \mathcal{P}_J^{-1} , \mathcal{P}_{GSL}^{-1} , \mathcal{P}_{DU}^{-1} , and \mathcal{P}_{LD}^{-1} applied to a 2D heat equation with s -stage Radau IIA methods. Here, $h_x = 2^{-3}$ and $h_t = h_x^{\frac{p+1}{2s-1}}$, where $p = 2$ is the degree of the Lagrange polynomial basis functions in space. Preconditioners are constructed exactly.

s	$\kappa(\mathcal{A})$	$\kappa(\mathcal{AP}_J^{-1})$	$\kappa(\mathcal{AP}_{GSL}^{-1})$	$\kappa(\mathcal{AP}_{DU}^{-1})$	$\kappa(\mathcal{AP}_{LD}^{-1})$
2	240.37	3.23	1.75	5.32	2.48
3	502.53	5.66	2.58	11.18	2.66
4	746.23	8.54	3.63	18.23	3.04
5	959.16	11.76	5.08	26.53	3.21
6	1137.24	15.23	7.13	35.97	3.50
7	1281.47	18.90	10.05	46.48	3.67

In Table 4.2, we examine the 2-norm condition number for the unpreconditioned system \mathcal{A} arising from the double-glazing problem, for \mathcal{A} preconditioned on the left by \mathcal{P}_{GSL} and \mathcal{P}_{LD} , and for \mathcal{A} preconditioned on the right by \mathcal{P}_{GSL} and \mathcal{P}_{LD} . We

have set $\epsilon = 0.005$, which corresponds to a weakly advection-dominated problem. We are using an s -stage Radau IIA IRK method for s ranging from 2 to 7. In all of these results, we have chosen $h_x = 2^{-4}$ and $h_t = h_x^{\frac{p+1}{2s-1}}$, where $p = 1$ is the degree of the Lagrange polynomial basis functions in space. The preconditioners are constructed exactly. As with the heat problem, both of the preconditioners significantly reduce the condition number. In this table, we omit results for the block Jacobi preconditioner and our upper triangular preconditioner since, as in the heat test problem, these did not perform as well as \mathcal{P}_{GSL} and \mathcal{P}_{LD} . The \mathcal{P}_{LD} preconditioner applied on the right gives greater improvement than \mathcal{P}_{GSL} applied on the right for the higher-order stages $s = 4$ through $s = 7$. \mathcal{P}_{LD} applied on the left gives the greatest improvement in condition number for all stages.

Table 4.2: Condition numbers of left-preconditioned and right-preconditioned matrices with preconditioners \mathcal{P}_{GSL}^{-1} and \mathcal{P}_{LD}^{-1} applied to a 2D double-glazing problem with $\epsilon = 0.005$, and with s -stage Radau IIA methods. Here, $h_x = 2^{-4}$ and $h_t = h_x^{\frac{p+1}{2s-1}}$, where $p = 1$ is the degree of the Lagrange polynomial basis functions in space. Preconditioners are constructed exactly.

s	$\kappa(\mathcal{A})$	$\kappa(\mathcal{P}^{-1}\mathcal{A})$		$\kappa(\mathcal{A}\mathcal{P}^{-1})$	
		$\kappa(\mathcal{P}_{GSL}^{-1}\mathcal{A})$	$\kappa(\mathcal{P}_{LD}^{-1}\mathcal{A})$	$\kappa(\mathcal{A}\mathcal{P}_{GSL}^{-1})$	$\kappa(\mathcal{A}\mathcal{P}_{LD}^{-1})$
2	631.32	1.67	1.26	1.73	2.40
3	1237.60	2.67	1.53	2.51	2.57
4	1802.02	4.14	1.80	3.54	2.92
5	2310.65	6.47	2.05	4.92	3.07
6	2762.17	10.23	2.25	6.83	3.32
7	3155.26	16.30	2.44	9.55	3.46

With GMRES, the condition number does not always tell the whole story, and it is also important to consider how well the preconditioner clusters the eigenvalues. In Figures 4.1 and 4.2, we show eigenvalues plotted in the complex plane for the matrix \mathcal{A} and preconditioned systems. In Figure 4.1 the matrix \mathcal{A} is from the 2D heat problem and this system is right preconditioned with \mathcal{P}_J , \mathcal{P}_{GSL} , \mathcal{P}_{DU} , and \mathcal{P}_{LD} . In Figure 4.2, \mathcal{A} is from the 2D double-glazing problem with $\epsilon = 0.005$ and this system is left preconditioned with \mathcal{P}_{GSL} and \mathcal{P}_{LD} . In both figures, we are using an s -stage Radau IIA IRK method for s ranging from 2 to 5, and we have again chosen $h_x = 2^{-3}$ and $h_t = h_x^{\frac{p+1}{2s-1}}$, where p is the degree of the Lagrange polynomial basis functions in space; $p = 2$ for the heat problem and $p = 1$ for the double-glazing problem. The preconditioners are constructed exactly. In both cases, the original matrix \mathcal{A} has a number of eigenvalues very near zero. All of the preconditioners succeed in clustering the eigenvalues farther away from zero. In the heat problem with $s = 2$, \mathcal{P}_{GSL} and \mathcal{P}_{LD} give the tightest clustering and have very similar eigenvalues. But as s increases, \mathcal{P}_{LD} maintains tighter clustering than \mathcal{P}_{GSL} . This fact is reflected in our numerical results presented in the next section. In the double-glazing problem, the original matrix \mathcal{A} also has a number of eigenvalues very near zero. In this problem, both preconditioners cluster the eigenvalues farther from zero. Our \mathcal{P}_{LD} preconditioner gives the tightest clustering for all stages s .

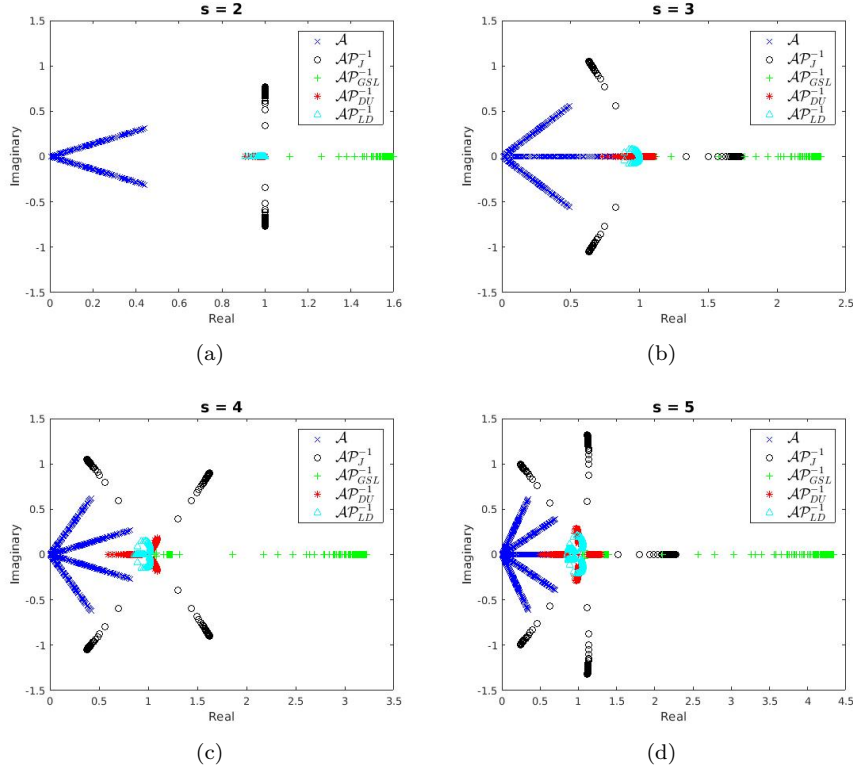


Fig. 4.1: Eigenvalues of the matrices \mathcal{A} , \mathcal{AP}_J^{-1} , \mathcal{AP}_{GSL}^{-1} , \mathcal{AP}_{DU}^{-1} , and \mathcal{AP}_{LD}^{-1} for 2D heat problem with Radau IIA $s = 2$ (a), $s = 3$ (b), $s = 4$ (c), and $s = 5$ (d). The x -axis is the real part and the y -axis is the imaginary part of the eigenvalue. Preconditioners are constructed exactly.

4.4. Comparison between \mathcal{P}_{GSL} with optimal coefficients and \mathcal{P}_{LD} preconditioners. In G.A. Staff et al. [7], the authors develop an optimal block lower triangular Gauss–Seidel preconditioner. This block Gauss–Seidel preconditioner is derived such that the preconditioner coefficients \tilde{A}_{GSL} are computed from the following optimization problem

$$(4.4) \quad \begin{aligned} \min_{\tilde{A}_{GSL}} \quad & \kappa(\tilde{A}_{GSL}^{-1}A) \\ \text{s.t.} \quad & \text{diag}(\tilde{A}_{GSL}) = \text{diag}(A). \end{aligned}$$

That is, their optimal preconditioner, which we will refer to as $\tilde{\mathcal{P}}_{GSL}$, uses coefficients that optimize the condition number of the (left) preconditioned system for a block lower triangular matrix subject to the given constraint.

In Table 4.3, we examine the 2-norm condition number for \mathcal{A} left preconditioned by $\tilde{\mathcal{P}}_{GSL}$ and \mathcal{P}_{LD} for 1D and 2D heat problems. For $\tilde{\mathcal{P}}_{GSL}$, we use the optimized \tilde{A}_{GSL} coefficients given in [7]. We fix $h_x = 2^{-3}$ and $h_t = h_x^{\frac{p+1}{q}}$, where $p = 2$ is the degree of the Lagrange polynomial basis functions in space and q is the order of the corresponding IRK method (that is, $q = 2s - 1$ for Radau IIA and $q = 2s - 2$ for

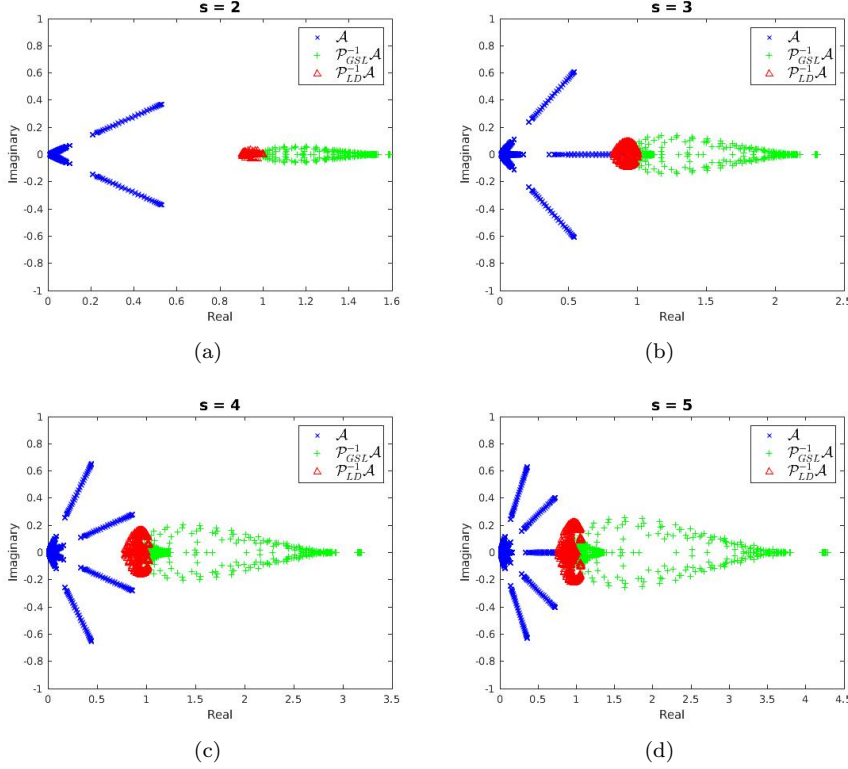


Fig. 4.2: Eigenvalues of the matrices \mathcal{A} , $\mathcal{P}_{GSL}^{-1}\mathcal{A}$, and $\mathcal{P}_{LD}^{-1}\mathcal{A}$ for 2D double-glazing problem with $\epsilon = 0.005$, and with Radau IIA $s = 2$ (a), $s = 3$ (b), $s = 4$ (c), and $s = 5$ (d). The x -axis is the real part and the y -axis is the imaginary part of the eigenvalue. Preconditioners are constructed exactly.

Lobatto IIIC'). We use Radau IIA with $s = 2$ through $s = 6$ and Lobatto IIIC with $s = 2$ through $s = 4$. For the Lobatto IIIC problems, we see that $\tilde{\mathcal{P}}_{GSL}$ consistently yields a lower condition number than \mathcal{P}_{LD} for all stages s . For the Radau IIA problems, however, the \mathcal{P}_{LD} preconditioner yields a lower condition number than the optimized preconditioner \mathcal{P}_{GSL} for all stages s . We note that in [7], the authors comment that they may not obtain the global minimum to (4.4) with the optimization process they use. The values used in \mathcal{P}_{LD} give a lower condition number.

Again we note that for GMRES, a lower condition number does not always indicate a superior preconditioner. In Figures 4.3 and 4.4 we see that \mathcal{P}_{LD} more effectively clusters the eigenvalues for both Radau IIA and Lobatto IIIC for stages $s = 2, 3, 4$ and 5.

5. Numerical Experiments. To test the performance of our preconditioners we consider the two model problems described in Section 3: a 2D heat equation problem and the 2D advection-diffusion double-glazing problem. For both we use s -stage Radau IIA or Lobatto IIIC methods for time discretization. For the heat equation we discretize in space using Galerkin finite elements with piecewise quadratic basis functions on triangular meshes. For the double-glazing problem we use streamwise

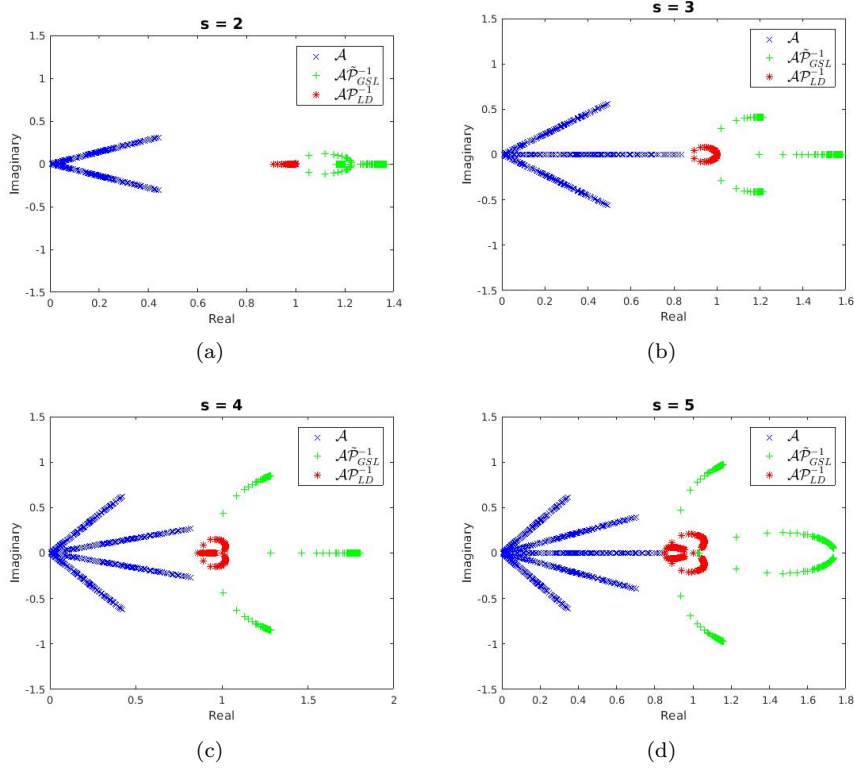


Fig. 4.3: Eigenvalues of the matrix \mathcal{A} , $\tilde{\mathcal{A}}\tilde{\mathcal{P}}_{GSL}^{-1}$, and $\mathcal{A}\mathcal{P}_{LD}^{-1}$ for 2D heat problem with Radau IIA $s = 2$ (a), $s = 3$ (b), $s = 4$ (c), and $s = 5$ (d). The x -axis is the real part and the y -axis is the imaginary part of the eigenvalue. Preconditioners are constructed exactly.

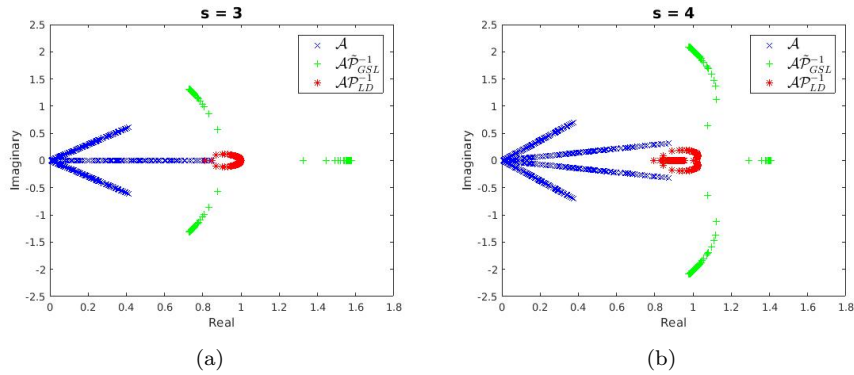


Fig. 4.4: Eigenvalues of the matrix \mathcal{A} , $\tilde{\mathcal{A}}\tilde{\mathcal{P}}_{GSL}^{-1}$, and $\mathcal{A}\mathcal{P}_{LD}^{-1}$ for 2D heat problem with Lobatto IIIC stages $s = 3$ (a) and $s = 4$ (b). The x -axis is the real part and the y -axis is the imaginary part of the eigenvalue. Preconditioners are constructed exactly.

Table 4.3: Condition numbers of the left-preconditioned system with preconditioners $\tilde{\mathcal{P}}_{GSL}$ (\mathcal{P}_{GSL} with optimal coefficients) and \mathcal{P}_{LD} for various IRK methods applied to 1D and 2D heat problems. Here, $h_x = 2^{-3}$ and $h_t = h_x^{\frac{p+1}{q}}$, where $p = 2$ is the degree of the Lagrange polynomial in space and q is the order of the corresponding IRK method (that is, $q = 2s - 1$ for Radau IIA and $q = 2s - 2$ for Lobatto IIIC). Preconditioners are constructed exactly.

	1D		2D	
	$\kappa(\tilde{\mathcal{P}}_{GSL}^{-1}\mathcal{A})$	$\kappa(\mathcal{P}_{LD}^{-1}\mathcal{A})$	$\kappa(\tilde{\mathcal{P}}_{GSL}^{-1}\mathcal{A})$	$\kappa(\mathcal{P}_{LD}^{-1}\mathcal{A})$
R IIA 2	1.58	1.26	1.60	1.26
R IIA 3	1.94	1.50	1.93	1.52
R IIA 4	2.18	1.74	2.17	1.77
R IIA 5	2.37	1.95	2.40	1.98
R IIA 6	3.01	2.14	3.18	2.18
L IIIC 2	1.48	2.68	1.54	2.78
L IIIC 3	3.20	6.68	3.32	6.94
L IIIC 4	4.92	10.43	5.24	10.93

upwind Petrov-Galerkin discretization [1] with piecewise linear basis functions, again on triangular meshes. Dirichlet boundary conditions are imposed using Nitsche's method [8, 3].

Production of the finite element matrices is carried out with the Sundance finite element toolkit [4]. The method of manufactured solutions is used to produce an exact solution to each linear subproblem so that we can check the error in addition to the residual.

A common practice in analysis of methods for spatiotemporal problems is to choose $h_t \sim h_x$ and increase the polynomial order of spatial discretization in lockstep with the order of time discretization. Here, we use a different approach. We use low-order elements ($p = 1$ or $p = 2$) throughout, and then for each h_x we choose a timestep h_t such that the spatial discretization error and temporal global truncation error are comparable. With a Radau s -stage method, we set $h_x^{p+1} = h_t^{2s-1}$ so that the timestep is $h_t = h_x^{\frac{p+1}{2s-1}}$, and with a Lobatto IIIC method, we set $h_x^{p+1} = h_t^{2s-2}$ so that the timestep is $h_t = h_x^{\frac{p+1}{2s-2}}$. By holding the spatial order p fixed, we are simulating conditions in which a modeler uses a high order integrator to obtain results of a specified accuracy with larger timestep and smaller computational cost.

We solve the linear system (3.2) using preconditioned GMRES. We construct our preconditioners as described in Section 4 using one AMG V-cycle for each subsolve. We use AMG from the IFISS software package [6]. When applied to the advection-diffusion equation, multilevel algorithms and smoothers must be handled with care [1]. In these experiments we use the strategy of ensuring that the SUPG discretization is stabilized on the coarse mesh, and use the smoothed aggregation smoother from IFISS. All results are computed in MATLAB on a machine with an Intel Core i7 1.80 GHz (Turbo Boost up to 4.00 GHz) processor and 8.00 GB RAM.

In Section 4, we examined condition numbers and eigenvalue distributions for the preconditioners \mathcal{P}_J , \mathcal{P}_{GSL} , $\tilde{\mathcal{P}}_{GSL}$, \mathcal{P}_{DU} , and \mathcal{P}_{LD} applied to the 2D heat equation

(3.1) and the double-glazing advection-diffusion problem (3.3). In this section, we investigate the convergence performance of our preconditioner compared to the others. We start with the 2D heat equation (3.1) in Section 5.1 and compare iteration counts and timing results for GMRES preconditioned on the right by \mathcal{P}_J , \mathcal{P}_{GSL} , \mathcal{P}_{DU} , and \mathcal{P}_{LD} for various problem sizes and varying orders of IRK methods. We then examine the robustness of the \mathcal{P}_{GSL} and \mathcal{P}_{LD} preconditioners for a fixed spatial resolution as we vary the time step size. We also compare our preconditioner with the optimized preconditioner $\tilde{\mathcal{P}}_{GSL}$. We conclude the heat equation results by examining the relative errors achieved with each of the preconditioners. In Section 5.2 we give iteration count and timing results for the best performing of the preconditioners, \mathcal{P}_{GSL} and \mathcal{P}_{LD} , applied to the double-glazing advection-diffusion problem (3.3). As with the heat equation results, we first compare methods for varying spatial resolutions and various order IRK methods. We then fix the spatial resolution and compare the robustness the methods as we vary the time step size.

5.1. Heat Equation Results. We first test the performance of the preconditioners \mathcal{P}_J , \mathcal{P}_{GSL} , \mathcal{P}_{DU} , and \mathcal{P}_{LD} , on our 2D heat problem (3.1).

5.1.1. Comparison results between \mathcal{P}_J , \mathcal{P}_{GSL} , \mathcal{P}_{DU} , and \mathcal{P}_{LD} preconditioners. In Table 5.1 we report iteration counts and timing results for right-preconditioned GMRES to converge with relative tolerance 1.0×10^{-8} for varying mesh sizes h_x and varying number of stages (varying order). Throughout, all of the triangular preconditioners outperform the block Jacobi preconditioner \mathcal{P}_J , as expected. The *LDU*-based block upper triangular preconditioner \mathcal{P}_{DU} and the block Gauss-Seidel preconditioner \mathcal{P}_{GSL} perform similarly, with \mathcal{P}_{GSL} performing slightly better, especially for the higher stage methods. However, overall, the *LDU*-based block lower triangular preconditioner \mathcal{P}_{LD} performs better than all of the other preconditioners in both iteration count and timing. For 2-stage Radau IIA, \mathcal{P}_{LD} and \mathcal{P}_{GSL} have similar performance, with \mathcal{P}_{LD} performing slightly better on the smaller problems and \mathcal{P}_{GSL} performing slightly better on the larger problems. But as the number of stages increases (that is, with increasing order), the performance of \mathcal{P}_{LD} improves over the other preconditioners giving consistently lower iteration counts and timing. For a given number of stages, all of the triangular preconditioners scale well with problem size, exhibiting little to no growth in iteration count with increasing problem size. However the \mathcal{P}_{LD} preconditioner has the least increase in cost as the number of stages increases.

We have also tested all four preconditioners for Lobatto IIIC methods. Table 5.2 shows iteration counts and timing results for right-preconditioned GMRES to converge with a relative tolerance of 1.0×10^{-8} for the 2D heat problem (3.1) using Lobatto IIIC methods in time and quadratic finite elements in space. For the Lobatto IIIC methods, the \mathcal{P}_{LD} preconditioner outperforms all of the other preconditioners both in iteration count and timing, with more significant improvements for larger problems and larger number of stages (higher order).

Because a common time-stepping solution method for (3.1) is an adaptive strategy where the time step increases as the steady-state is approached, we also investigate the effectiveness of our preconditioner for a fixed spatial resolution while varying the time step size. In Table 5.3, we report iteration counts and timing results for left and right-preconditioned GMRES to converge with a relative residual tolerance of 1.0×10^{-8} for the 2D heat problem with 2-stage and 7-stage Radau IIA methods. In this table, we have fixed the spatial resolution at $h_x^{-1} = 128$ and we vary the time step size h_t between 0.05 and 5.0. Both the \mathcal{P}_{GSL} preconditioner and our \mathcal{P}_{LD} preconditioner

Table 5.1: Iteration counts and elapsed time (times in seconds are shown in parentheses) for right-preconditioned GMRES to converge with relative residual tolerance 1.0×10^{-8} for a 2D heat problem with s -stage Radau IIA methods with preconditioners \mathcal{P}_J , \mathcal{P}_{GSL} , \mathcal{P}_{DU} , and \mathcal{P}_{LD} . Here we choose $h_t = h_x^{\frac{p+1}{2s-1}}$, where $p = 2$ is the degree of the Lagrange polynomial in space. Preconditioners are approximated using one AMG V-cycle for each subsolve.

stage	h_x^{-1}	DOF	\mathcal{P}_J	\mathcal{P}_{GSL}	\mathcal{P}_{DU}	\mathcal{P}_{LD}
s=2	8	450	14 (0.03)	8 (0.01)	7 (0.01)	7 (0.01)
	16	1922	14 (0.05)	8 (0.02)	7 (0.02)	7 (0.02)
	32	7938	14 (0.10)	8 (0.06)	7 (0.06)	7 (0.05)
	64	32,258	14 (0.40)	7 (0.19)	7 (0.19)	7 (0.19)
	128	130,050	14 (1.48)	7 (0.80)	7 (0.83)	7 (0.82)
s=3	8	675	23 (0.06)	10 (0.05)	10 (0.03)	9 (0.03)
	16	2883	22 (0.06)	10 (0.03)	10 (0.03)	8 (0.02)
	32	11,907	21 (0.17)	10 (0.09)	10 (0.08)	8 (0.07)
	64	48,387	22 (0.67)	9 (0.29)	10 (0.32)	8 (0.26)
	128	195,075	21 (2.90)	8 (1.12)	9 (1.25)	8 (1.12)
s=4	8	900	32 (0.10)	13 (0.08)	13 (0.07)	10 (0.04)
	16	3844	31 (0.11)	12 (0.05)	13 (0.05)	10 (0.04)
	32	15,876	30 (0.31)	12 (0.13)	12 (0.13)	10 (0.12)
	64	64,516	29 (1.16)	11 (0.47)	12 (0.52)	9 (0.39)
	128	260,100	27 (5.24)	11 (2.10)	12 (2.29)	9 (1.73)
s=5	8	1125	42 (0.10)	15 (0.08)	16 (0.08)	11 (0.06)
	16	4805	40 (0.23)	15 (0.08)	16 (0.08)	11 (0.06)
	32	19,845	39 (0.50)	13 (0.18)	15 (0.20)	11 (0.17)
	64	80,645	35 (1.80)	13 (0.70)	15 (0.81)	11 (0.61)
	128	325,125	33 (8.44)	12 (2.98)	14 (3.49)	11 (2.75)
s=6	8	1350	53 (0.15)	18 (0.08)	19 (0.08)	12 (0.04)
	16	5766	50 (0.31)	17 (0.11)	18 (0.11)	12 (0.08)
	32	23,814	46 (0.73)	16 (0.27)	18 (0.29)	12 (0.19)
	64	96,774	42 (2.66)	15 (1.02)	17 (1.14)	12 (0.81)
	128	390,150	38 (12.13)	14 (4.27)	17 (5.41)	12 (3.56)
s=7	8	1575	62 (0.22)	21 (0.10)	23 (0.10)	13 (0.09)
	16	6727	57 (0.40)	20 (0.15)	22 (0.16)	13 (0.09)
	32	27,783	52 (0.98)	19 (0.37)	21 (0.40)	13 (0.26)
	64	112,903	48 (3.64)	18 (1.49)	20 (1.61)	12 (0.98)
	128	455,175	44 (17.06)	17 (6.29)	20 (7.24)	12 (4.42)

are quite robust with respect to varying time step size. Both preconditioners perform very similarly (and robustly) for the lower-order $s = 2$ method; for the higher-order $s = 7$ method, both preconditioners are robust with respect to the varying time step size, but our \mathcal{P}_{LD} preconditioner outperforms \mathcal{P}_{GSL} as a left preconditioner and as a right preconditioner for all values of the time step.

Table 5.2: Iteration counts and elapsed time (times in seconds are shown in parentheses) for right-preconditioned GMRES to converge with relative residual tolerance 1.0×10^{-8} for a 2D heat problem with s -stage Lobatto IIIC methods with preconditioners \mathcal{P}_J , \mathcal{P}_{GSL} , \mathcal{P}_{DU} , and \mathcal{P}_{LD} . Here we choose $h_t = h_x^{\frac{p+1}{2s-2}}$, where $p = 2$ is the degree of the Lagrange polynomial in space. Preconditioners are approximated using one AMG V-cycle for each subsolve.

stage	h_x^{-1}	DOF	\mathcal{P}_J	\mathcal{P}_{GSL}	\mathcal{P}_{DU}	\mathcal{P}_{LD}
s=2	8	450	17 (0.01)	10 (0.01)	7 (0.03)	7 (0.01)
	16	1922	18 (0.03)	10 (0.02)	7 (0.02)	8 (0.02)
	32	7938	19 (0.13)	10 (0.07)	8 (0.06)	8 (0.06)
	64	32,258	19 (0.47)	10 (0.26)	7 (0.19)	8 (0.22)
	128	130,050	18 (2.19)	10 (1.21)	7 (0.86)	8 (1.00)
s=3	8	675	27 (0.05)	12 (0.03)	11 (0.02)	10 (0.02)
	16	2883	29 (0.07)	12 (0.03)	11 (0.03)	10 (0.03)
	32	11,907	27 (0.22)	12 (0.10)	11 (0.10)	10 (0.08)
	64	48,387	29 (0.96)	11 (0.36)	11 (0.34)	10 (0.35)
	128	195,075	27 (3.89)	11 (1.54)	10 (1.41)	9 (1.27)
s=4	8	900	40 (0.07)	16 (0.05)	15 (0.05)	12 (0.05)
	16	3844	40 (0.14)	15 (0.06)	15 (0.06)	12 (0.05)
	32	15,876	39 (0.41)	14 (0.15)	14 (0.15)	12 (0.13)
	64	64,516	38 (1.58)	14 (0.60)	13 (0.57)	11 (0.47)
	128	260,100	37 (7.58)	13 (2.46)	13 (2.44)	11 (2.06)
s=5	8	1125	55 (0.08)	19 (0.03)	19 (0.03)	13 (0.02)
	16	4805	55 (0.26)	18 (0.09)	18 (0.08)	13 (0.07)
	32	19,845	51 (0.57)	17 (0.20)	17 (0.22)	13 (0.16)
	64	80,645	49 (2.61)	16 (0.85)	16 (0.84)	12 (0.64)
	128	325,125	45 (12.01)	15 (3.64)	15 (3.69)	12 (2.96)

5.1.2. Comparison results between $\tilde{\mathcal{P}}_{GSL}$ with optimal coefficients and \mathcal{P}_{LD} preconditioners. In this section we compare our LDU -based block lower triangular preconditioner \mathcal{P}_{LD} with the optimal block lower triangular Gauss-Seidel preconditioner introduced in [7]. $\tilde{\mathcal{P}}_{GSL}$ was developed using coefficients optimized to reduce the condition number of the left-preconditioned system $\tilde{\mathcal{P}}_{GSL}^{-1}\mathcal{A}$.

In Table 5.4, we present results similar to those in Table 5.1. The problem set up is the same, but here we use left preconditioning for all of the preconditioners since the coefficients of $\tilde{\mathcal{P}}_{GSL}$ were optimized for left preconditioning, and we include an additional column for $\tilde{\mathcal{P}}_{GSL}$. The first thing we note is that although the idea of optimizing coefficients to reduce the condition number of the preconditioned system is a perfectly sensible idea, for this problem the unoptimized \mathcal{P}_{GSL} performs better than the optimized $\tilde{\mathcal{P}}_{GSL}$ except on the smallest problems. For all numbers of stages, \mathcal{P}_{GSL} performs better than $\tilde{\mathcal{P}}_{GSL}$ for problems sizes $h_x^{-1} = 32, 64$, and 128. We observe in this table that our \mathcal{P}_{LD} has the best performance overall. Although it performs slightly worse as a left preconditioner than it did in Table 5.1 as a right preconditioner, it nonetheless achieves the lowest iteration count and lowest timing of all of the preconditioners for all numbers of stages and all problem sizes.

Table 5.3: Iteration counts and elapsed time (times in seconds are shown in parentheses) for left-preconditioned and right-preconditioned GMRES to converge with preconditioned relative residual tolerance 1.0×10^{-8} for a 2D heat problem with s -stage Radau IIA methods with preconditioners \mathcal{P}_{GSL} and \mathcal{P}_{LD} . Here we keep $h_x^{-1} = 128$ fixed and vary h_t from 0.05 to 5.0. Preconditioners are approximated using one AMG V-cycle for each subsolve.

stage	h_t	left prec. GMRES		right prec. GMRES	
		\mathcal{P}_{GSL}	\mathcal{P}_{LD}	\mathcal{P}_{GSL}	\mathcal{P}_{LD}
$s = 2$	0.05	8 (0.85)	7 (0.71)	7 (0.63)	7 (0.63)
	0.1	8 (0.78)	7 (0.73)	6 (0.55)	7 (0.63)
	0.5	7 (0.74)	7 (0.71)	6 (0.56)	7 (0.63)
	1.0	7 (0.70)	7 (0.70)	6 (0.56)	7 (0.64)
	5.0	7 (0.72)	8 (0.78)	7 (0.63)	7 (0.65)
$s = 7$	0.05	24 (7.38)	14 (4.24)	19 (5.43)	12 (3.49)
	0.1	24 (7.76)	14 (4.53)	18 (5.62)	12 (3.70)
	0.5	23 (7.42)	14 (4.49)	17 (5.34)	13 (4.05)
	1.0	23 (6.77)	14 (4.56)	17 (4.89)	13 (4.00)
	5.0	21 (6.74)	15 (4.93)	18 (5.64)	13 (4.17)

In Table 5.5, we present results similar to those in Table 5.4 but using Lobatto IIIC methods. The problem set up is again the same, using left preconditioning for all of the preconditioners. The results are very similar to those in the previous table. The unoptimized \mathcal{P}_{GSL} performs better than the optimized $\tilde{\mathcal{P}}_{GSL}$ on all of these problems. Our \mathcal{P}_{LD} preconditioner has the best performance overall.

It should be noted that since the results in Tables 5.4 and 5.5 are using left preconditioning, the relative residuals are actually preconditioned relative residuals. So the norms of the various preconditioners could be having an effect on the iteration counts. Since all of our numerical tests were run using the method of manufactured solutions, in Tables 5.6 and 5.7 we examine the relative errors for each method.

In Table 5.6 we report the relative errors for various numbers of stages for Radau IIA and Lobatto IIIC with the five preconditioners \mathcal{P}_J , \mathcal{P}_{GSL} , $\tilde{\mathcal{P}}_{GSL}$, \mathcal{P}_{DU} , and \mathcal{P}_{LD} all used as left preconditioners. The problem size is fixed at $h_x^{-1} = 128$, which is the largest of the problems we considered, and $h_t = h_x^{\frac{p+1}{q}}$, where $p = 2$ is the degree of the Lagrange polynomial in space and q is the order of the corresponding IRK method (that is, $q = 2s - 1$ for Radau IIA and $q = 2s - 2$ for Lobatto IIIC). The stopping criteria is the same as in all previous tables. With left preconditioning, problems with Radau IIA and Lobatto IIIC methods and all numbers of stages yielded very similar relative errors.

In Table 5.7 we report the relative errors for various numbers of stages for Radau IIA and Lobatto IIIC with the four preconditioners \mathcal{P}_J , \mathcal{P}_{GSL} , \mathcal{P}_{DU} , and \mathcal{P}_{LD} this time all used as right preconditioners. We do not include the optimized GSL-based preconditioner $\tilde{\mathcal{P}}_{GSL}$ since it is optimized for left preconditioning. The problem size is again fixed at $h_x^{-1} = 128$, and as before $h_t = h_x^{\frac{p+1}{q}}$, where $p = 2$ is the degree of the Lagrange polynomial in space and q is the order of the corresponding IRK method

Table 5.4: Iteration counts and elapsed time (times in seconds are shown in parentheses) for left-preconditioned GMRES to converge with preconditioned relative residual tolerance 1.0×10^{-8} for a 2D heat problem with s -stage Radau IIA methods with preconditioners \mathcal{P}_J , \mathcal{P}_{GSL} , $\tilde{\mathcal{P}}_{GSL}$, \mathcal{P}_{DU} , and \mathcal{P}_{LD} . Here we choose $h_t = h_x^{\frac{p+1}{2s-1}}$, where $p = 2$ is the degree of the Lagrange polynomial in space. Preconditioners are approximated using one AMG V-cycle for each subsolve.

stage	h_x^{-1}	DOF	\mathcal{P}_J	\mathcal{P}_{GSL}	$\tilde{\mathcal{P}}_{GSL}$	\mathcal{P}_{DU}	\mathcal{P}_{LD}
s=2	8	450	14 (0.02)	9 (0.02)	9 (0.01)	7 (0.01)	7 (0.01)
	16	1922	16 (0.03)	9 (0.02)	10 (0.02)	8 (0.02)	7 (0.02)
	32	7938	16 (0.10)	9 (0.06)	10 (0.06)	8 (0.05)	7 (0.05)
	64	32,258	17 (0.38)	9 (0.24)	10 (0.27)	8 (0.23)	7 (0.21)
	128	130,050	18 (2.01)	9 (1.16)	10 (1.24)	8 (1.03)	7 (0.94)
s=3	8	675	23 (0.06)	11 (0.04)	13 (0.03)	10 (0.04)	9 (0.03)
	16	2883	25 (0.05)	12 (0.03)	13 (0.03)	11 (0.03)	9 (0.02)
	32	11,907	25 (0.20)	12 (0.09)	14 (0.10)	11 (0.08)	9 (0.07)
	64	48,387	27 (0.77)	12 (0.36)	14 (0.43)	11 (0.34)	9 (0.31)
	128	195,075	28 (3.65)	12 (1.65)	14 (1.91)	11 (1.56)	9 (1.32)
s=4	8	900	32 (0.07)	15 (0.06)	18 (0.05)	14 (0.06)	11 (0.04)
	16	3844	34 (0.13)	15 (0.06)	19 (0.07)	14 (0.06)	11 (0.05)
	32	15,876	36 (0.38)	15 (0.17)	19 (0.21)	14 (0.16)	11 (0.13)
	64	64,516	36 (1.52)	15 (0.70)	19 (0.84)	14 (0.64)	11 (0.51)
	128	260,100	36 (6.51)	15 (2.88)	19 (3.63)	14 (2.74)	11 (2.20)
s=5	8	1125	44 (0.11)	18 (0.08)	22 (0.06)	17 (0.07)	12 (0.06)
	16	4805	45 (0.21)	18 (0.08)	22 (0.10)	17 (0.08)	12 (0.06)
	32	19,845	46 (0.61)	18 (0.26)	22 (0.28)	17 (0.23)	12 (0.17)
	64	80,645	47 (2.21)	18 (0.90)	22 (1.18)	17 (0.93)	12 (0.74)
	128	325,125	47 (11.15)	18 (4.41)	22 (5.45)	17 (4.26)	12 (3.00)
s=6	8	1350	55 (0.14)	22 (0.08)	24 (0.07)	20 (0.07)	13 (0.04)
	16	5766	55 (0.32)	22 (0.14)	24 (0.16)	21 (0.14)	14 (0.09)
	32	23,814	56 (0.93)	22 (0.38)	24 (0.40)	20 (0.34)	14 (0.24)
	64	96,774	56 (3.77)	21 (1.49)	24 (1.69)	20 (1.42)	13 (0.94)
	128	390,150	56 (15.89)	21 (6.21)	23 (6.89)	20 (5.63)	13 (3.98)

(that is, $q = 2s - 1$ for Radau IIA and $q = 2s - 2$ for Lobatto IIIC). The stopping criteria is the same as in all previous tables. With right reconditioning, our \mathcal{P}_{LD} preconditioner achieves almost an order of magnitude better relative error on almost all of the runs.

5.2. Advection-Diffusion Equation Results. In this section, we examine the performance of the preconditioners \mathcal{P}_{GSL} and \mathcal{P}_{LD} , on the 2D double-glazing advection-diffusion problem (3.3). For the double-glazing problem, we only present results for the \mathcal{P}_{GSL} and \mathcal{P}_{LD} preconditioners since in all of our previous experiments they performed best. We apply both as left preconditioners in these results, but our results are similar when they are applied as right preconditioners.

In Table 5.8, we report iteration counts and elapsed time for left-preconditioned

Table 5.5: Iteration counts and elapsed time (times in seconds are shown in parentheses) for left-preconditioned GMRES to converge with preconditioned relative residual tolerance 1.0×10^{-8} for a 2D heat problem with s -stage Lobatto IIIC methods with preconditioners \mathcal{P}_J , \mathcal{P}_{GSL} , $\tilde{\mathcal{P}}_{GSL}$, \mathcal{P}_{DU} , and \mathcal{P}_{LD} . Here we choose $h_t = h_x^{\frac{p+1}{2s-2}}$, where $p = 2$ is the degree of the Lagrange polynomial in space. Preconditioners are approximated using one AMG V-cycle for each subsolve.

stage	h_x^{-1}	DOF	\mathcal{P}_J	\mathcal{P}_{GSL}	$\tilde{\mathcal{P}}_{GSL}$	\mathcal{P}_{DU}	\mathcal{P}_{LD}
s=2	8	450	18 (0.02)	11 (0.02)	18 (0.02)	7 (0.02)	8 (0.02)
	16	1922	20 (0.04)	12 (0.02)	20 (0.03)	8 (0.02)	8 (0.02)
	32	7938	21 (0.15)	12 (0.09)	21 (0.14)	8 (0.06)	8 (0.06)
	64	32,258	21 (0.56)	12 (0.34)	21 (0.56)	8 (0.24)	8 (0.23)
	128	130,050	21 (2.40)	12 (1.49)	21 (2.50)	8 (1.05)	8 (1.05)
s=3	8	675	27 (0.06)	14 (0.04)	25 (0.04)	11 (0.03)	11 (0.03)
	16	2883	32 (0.08)	14 (0.04)	27 (0.06)	12 (0.03)	11 (0.03)
	32	11,907	33 (0.23)	15 (0.11)	28 (0.20)	12 (0.09)	11 (0.09)
	64	48,387	35 (1.06)	15 (0.49)	30 (0.98)	12 (0.40)	11 (0.37)
	128	195,075	37 (3.89)	15 (1.87)	30 (3.76)	12 (1.67)	11 (1.58)
s=4	8	900	41 (0.09)	18 (0.06)	31 (0.07)	15 (0.06)	13 (0.05)
	16	3844	44 (0.13)	18 (0.06)	35 (0.11)	16 (0.06)	13 (0.05)
	32	15,876	46 (0.51)	18 (0.20)	35 (0.38)	16 (0.18)	13 (0.15)
	64	64,516	49 (2.14)	18 (0.80)	37 (1.66)	16 (0.74)	13 (0.60)
	128	260,100	49 (8.93)	18 (3.23)	36 (6.60)	16 (3.11)	13 (2.53)

Table 5.6: Relative error for left-preconditioned GMRES converging to a preconditioned relative residual tolerance of 1.0×10^{-8} for a 2D heat problem with various IRK methods with preconditioners \mathcal{P}_J , \mathcal{P}_{GSL} , $\tilde{\mathcal{P}}_{GSL}$, \mathcal{P}_{DU} , and \mathcal{P}_{LD} . Here, $h_x = 2^{-7}$ and $h_t = h_x^{\frac{p+1}{q}}$, where $p = 2$ is the degree of the Lagrange polynomial in space and q is the order of the corresponding IRK method (that is, $q = 2s - 1$ for Radau IIA and $q = 2s - 2$ for Lobatto IIIC). Preconditioners are approximated using one AMG V-cycle for each subsolve.

IRK method	\mathcal{P}_J	\mathcal{P}_{GSL}	$\tilde{\mathcal{P}}_{GSL}$	\mathcal{P}_{DU}	\mathcal{P}_{LD}
R IIA 2	7.1×10^{-9}	6.1×10^{-9}	1.4×10^{-9}	1.8×10^{-9}	4.0×10^{-9}
R IIA 3	8.5×10^{-9}	6.9×10^{-9}	7.9×10^{-9}	9.4×10^{-9}	6.4×10^{-9}
R IIA 4	4.6×10^{-8}	8.5×10^{-9}	9.1×10^{-9}	3.8×10^{-8}	2.9×10^{-9}
R IIA 5	5.1×10^{-8}	9.0×10^{-9}	8.1×10^{-9}	4.6×10^{-8}	5.4×10^{-9}
R IIA 6	7.7×10^{-8}	1.0×10^{-8}	8.3×10^{-9}	9.8×10^{-8}	7.9×10^{-9}
L IIIC 2	8.0×10^{-9}	1.8×10^{-9}	8.0×10^{-9}	3.9×10^{-9}	6.6×10^{-9}
L IIIC 3	2.0×10^{-8}	6.7×10^{-9}	1.0×10^{-8}	1.8×10^{-8}	1.1×10^{-8}
L IIIC 4	5.7×10^{-8}	1.2×10^{-8}	7.9×10^{-9}	2.3×10^{-8}	8.9×10^{-9}

Table 5.7: Relative error for right-preconditioned GMRES converging to a relative residual tolerance of 1.0×10^{-8} for a 2D heat problem with various IRK methods with preconditioners \mathcal{P}_J , \mathcal{P}_{GSL} , \mathcal{P}_{DU} , and \mathcal{P}_{LD} . Here, $h_x = 2^{-7}$ and $h_t = h_x^{\frac{p+1}{q}}$, where $p = 2$ is the degree of the Lagrange polynomial in space and q is the order of the corresponding IRK method (that is, $q = 2s - 1$ for Radau IIA and $q = 2s - 2$ for Lobatto IIIC). Preconditioners are approximated using one AMG V-cycle for each subsolve.

IRK method	\mathcal{P}_J	\mathcal{P}_{GSL}	\mathcal{P}_{DU}	\mathcal{P}_{LD}
R IIA 2	1.5×10^{-6}	1.4×10^{-6}	2.8×10^{-7}	3.1×10^{-8}
R IIA 3	1.5×10^{-5}	1.5×10^{-5}	9.0×10^{-6}	2.8×10^{-7}
R IIA 4	2.7×10^{-5}	4.6×10^{-6}	8.3×10^{-6}	6.9×10^{-7}
R IIA 5	3.8×10^{-5}	1.1×10^{-5}	1.5×10^{-5}	1.3×10^{-7}
R IIA 6	5.6×10^{-5}	2.0×10^{-5}	9.5×10^{-6}	8.2×10^{-8}
R IIA 7	1.7×10^{-6}	3.9×10^{-7}	8.3×10^{-7}	2.4×10^{-7}
L IIIC 2	2.6×10^{-7}	1.3×10^{-7}	2.5×10^{-7}	2.1×10^{-8}
L IIIC 3	1.2×10^{-5}	8.1×10^{-6}	9.2×10^{-6}	6.9×10^{-6}
L IIIC 4	2.5×10^{-5}	1.3×10^{-5}	1.6×10^{-5}	1.4×10^{-6}
L IIIC 5	6.1×10^{-5}	1.3×10^{-5}	4.0×10^{-5}	1.2×10^{-6}

GMRES to converge with preconditioned relative residual tolerance 1.0×10^{-8} for the double-glazing problem (3.3) with s -stage Radau IIA methods using the preconditioners \mathcal{P}_{GSL} and \mathcal{P}_{LD} . We vary the number of stages from $s = 2$ to $s = 7$ for problems with spatial resolution $h_x^{-1} = 64$ and 128. Here we choose $h_t = h_x^{\frac{p+1}{2s-1}}$, where $p = 1$ is the degree of the Lagrange polynomial in space. We present results for $\epsilon = 0.04$, which is diffusion dominated and $\epsilon = 0.005$, which is weakly advection dominated. Preconditioners are approximated using one AMG V-cycle for each subsolve.

For the lowest-order problems with $s = 2$, \mathcal{P}_{GSL} and our \mathcal{P}_{LD} preconditioner perform similarly. For the diffusion dominated problem, $\epsilon = 0.04$, \mathcal{P}_{LD} outperforms \mathcal{P}_{GSL} for all of the higher-order problems, $s = 3$ through $s = 7$, with the improved performance being more pronounced on the highest-order problems. For the weakly advection-dominated problem, $\epsilon = 0.005$, the results are more mixed. For the smaller problems of each order, $h_x^{-1} = 64$, \mathcal{P}_{LD} takes fewer iterations in most cases, but takes longer in compute time in a few cases. For the largest higher-order problems, $s = 4$ through $s = 7$ with $h_x^{-1} = 128$, \mathcal{P}_{LD} performs better than \mathcal{P}_{GSL} in both iteration count and timing.

As we did for the heat equation, in Table 5.9, we investigate the robustness of the preconditioners for a fixed spatial resolution while varying the time step size. In this table, we present iteration counts and timing results for left-preconditioned GMRES to converge with a preconditioned relative residual tolerance of 1.0×10^{-8} for the 2D double-glazing problem with s -stage Radau IIA methods using the preconditioners \mathcal{P}_{GSL} and \mathcal{P}_{LD} . We present results for a lower-order method $s = 2$ and a higher-order method $s = 7$ and for $\epsilon = 0.04$, which is diffusion dominated, and $\epsilon = 0.005$, which is weakly advection dominated. In this table, we keep $h_x^{-1} = 128$ fixed and vary the time step size h_t from 0.05 to 5.0. Preconditioners are approximated using one AMG V-cycle for each subsolve.

Table 5.8: Iteration counts and elapsed time (times in seconds are shown in parentheses) for left-preconditioned GMRES to converge with preconditioned relative residual tolerance 1.0×10^{-8} for the double-glazing problem with s -stage Radau IIA methods with preconditioners \mathcal{P}_{GSL} and \mathcal{P}_{LD} . Here we choose $h_t = h_x^{\frac{p+1}{2s-1}}$, where $p = 1$ is the degree of the Lagrange polynomial in space. Preconditioners are approximated using one AMG V-cycle for each subsolve.

				$\epsilon = 0.04$		$\epsilon = 0.005$	
stage	h_x^{-1}	h_t	DOF	\mathcal{P}_{GSL}	\mathcal{P}_{LD}	\mathcal{P}_{GSL}	\mathcal{P}_{LD}
$s = 2$	64	0.0625	33282	9 (0.50)	7 (0.43)	11 (0.89)	11 (0.83)
	128	0.0394	132098	9 (1.72)	7 (1.44)	10 (3.74)	10 (3.84)
$s = 3$	64	0.1895	49923	14 (1.13)	10 (0.84)	17 (1.92)	16 (1.83)
	128	0.1436	198147	14 (3.96)	9 (2.73)	15 (8.23)	14 (9.71)
$s = 4$	64	0.3048	66564	18 (1.95)	12 (1.28)	22 (3.22)	21 (3.04)
	128	0.25	264196	18 (6.47)	11 (4.44)	20 (16.83)	17 (15.17)
$s = 5$	64	0.3969	83205	22 (2.62)	13 (1.58)	26 (4.54)	25 (4.29)
	128	0.3402	330245	22 (10.38)	13 (6.45)	25 (22.75)	21 (22.33)
$s = 6$	64	0.4695	99846	25 (3.94)	14 (2.33)	31 (6.34)	28 (6.51)
	128	0.4139	396294	27 (15.84)	14 (10.59)	30 (34.10)	23 (33.38)
$s = 7$	64	0.5274	116487	30 (5.39)	16 (2.81)	36 (8.53)	30 (9.04)
	128	0.4740	462343	31 (20.89)	15 (10.34)	36 (55.84)	25 (36.33)

For the diffusion-dominated problem with the lower-order method, our \mathcal{P}_{LD} preconditioner mildly outperforms \mathcal{P}_{GSL} for all time step sizes, and both preconditioners are robust with respect to varying the time step size. For the higher-order method, both preconditioners take more iterations and more time as the step size increases, but \mathcal{P}_{LD} grows less dramatically than \mathcal{P}_{GSL} and takes fewer iterations and less time for all time step sizes.

For the weakly advection-dominated problem, $\epsilon = 0.005$, \mathcal{P}_{LD} performs very similarly to \mathcal{P}_{GSL} on the lower-order problem, $s = 2$, and both preconditioners are robust with respect to the varying time step size. For the higher-order method, both preconditioners take more iterations and more time as the step size increases, but \mathcal{P}_{LD} grows much less dramatically than \mathcal{P}_{GSL} and takes fewer iterations and less time for all time step sizes.

6. Conclusion. In this paper, we have introduced a new preconditioner for the large, structured systems appearing in implicit Runge–Kutta time integration of parabolic PDE problems. Our preconditioner is based on a block LDU factorization, and for scalability, we have used a single AMG V-cycle on all subsolves. We have compared our preconditioner \mathcal{P}_{LD} in condition number and eigenvalue distribution to other preconditioners, and have demonstrated its effectiveness on two test problems, the heat equation and the double-glazing advection-diffusion problem. We have found that it is scalable (h_x -independent) and yields better timing results than other preconditioners currently in the literature: block Jacobi, block Gauss–Seidel, and the optimized block Gauss–Seidel method of [7]. It is also robust with respect to varying

Table 5.9: Iteration counts and elapsed time (times in seconds are shown in parentheses) for left-preconditioned GMRES to converge with preconditioned relative residual tolerance 1.0×10^{-8} for a 2D double-glazing problem with s -stage Radau IIA methods with preconditioners \mathcal{P}_{GSL} and \mathcal{P}_{LD} . Here we keep $h_x^{-1} = 128$ fixed and vary h_t from 0.05 to 5.0. Preconditioners are approximated using one AMG V-cycle for each subsolve.

stage	h_t	$\epsilon = 0.04$		$\epsilon = 0.005$	
		\mathcal{P}_{GSL}	\mathcal{P}_{LD}	\mathcal{P}_{GSL}	\mathcal{P}_{LD}
$s = 2$	0.05	9 (1.85)	7 (1.50)	11 (3.95)	10 (4.12)
	0.1	10 (1.96)	7 (1.43)	11 (4.24)	11 (5.01)
	0.5	11 (2.05)	8 (1.76)	12 (4.64)	11 (4.64)
	1.0	10 (2.08)	7 (1.47)	13 (4.76)	11 (4.72)
	5.0	10 (1.95)	8 (1.69)	12 (5.63)	12 (5.43)
$s = 7$	0.05	26 (18.39)	13 (9.78)	27 (41.80)	18 (28.56)
	0.1	27 (22.09)	14 (9.80)	28 (43.18)	21 (29.72)
	0.5	32 (24.97)	15 (12.29)	36 (53.60)	25 (40.76)
	1.0	37 (30.81)	17 (13.88)	45 (76.24)	26 (36.33)
	5.0	46 (35.33)	20 (16.14)	68 (109.26)	28 (36.66)

time step sizes for a fixed spatial resolution. We ran experiments with implicit Runge–Kutta stages up to $s = 7$, and have found that the new preconditioner outperforms the others, with the improvement becoming more pronounced as spatial discretization is refined and as temporal order is increased.

REFERENCES

- [1] H. ELMAN, D. SILVESTER, AND A. WATHEN, *Finite Elements and Fast Iterative Solvers*, Oxford University Press, 2005.
- [2] E. HAIRER, S. P. NØRSETT, AND G. WANNER, *Solving Ordinary Differential Equations I, Nonstiff Problems*, Springer Berlin Heidelberg, 1993.
- [3] M. JUNTUNEN AND R. STENBERG, *Nitsche’s method for general boundary conditions*, Mathematics of computation, 78 (2009), pp. 1353–1374.
- [4] K. LONG, R. KIRBY, AND B. VAN BLOEMEN WAANDERS, *Unified embedded parallel finite element computations via software-based Fréchet differentiation*, SIAM Journal on Scientific Computing, 32 (2010), pp. 3323–3351, <https://doi.org/10.1137/09076920X>, <http://epubs.siam.org/doi/10.1137/09076920X> (accessed 2019-07-29).
- [5] K.-A. MARDAL, T. K. NILSEN, AND G. A. STAFF, *Order-optimal preconditioners for implicit Runge–Kutta schemes applied to parabolic PDEs*, SIAM Journal on Scientific Computing, 29 (2007), pp. 361–375.
- [6] D. SILVESTER, H. ELMAN, AND A. RAMAGE, *Incompressible Flow and Iterative Solver Software (IFISS) version 3.5*, September 2016. <http://www.manchester.ac.uk/ifiss/>.
- [7] G. A. STAFF, K.-A. MARDAL, AND T. K. NILSEN, *Preconditioning of fully implicit Runge–Kutta schemes for parabolic PDEs*, Modeling, Identification and Control, 27 (2006), pp. 109–123, <https://doi.org/10.4173/mic.2006.2.3>.
- [8] R. STENBERG, *On some techniques for approximating boundary conditions in the finite element method*, Journal of Computational and applied Mathematics, 63 (1995), pp. 139–148.
- [9] G. WANNER AND E. HAIRER, *Solving Ordinary Differential Equations II: Stiff Problems*, Springer Berlin Heidelberg, 1996.