Robust and efficient preconditioners for the discontinuous Galerkin time-stepping method

IAIN SMEARS†
INRIA PARIS, 2 RUE SIMONE IFF, 75012 PARIS, FRANCE

The discontinuous Galerkin time-stepping method has many advantageous properties for solving parabolic equations. However, it requires the solution of a large nonsymmetric system at each time-step. This work develops a fully robust and efficient preconditioning strategy for solving these systems. Drawing on parabolic inf-sup theory, we first construct a left preconditioner that transforms the linear system to a symmetric positive definite problem to be solved by the preconditioned conjugate gradient algorithm. We then prove that the transformed system can be further preconditioned by an ideal block diagonal preconditioner, leading to a condition number bounded by 4 for any time-step size, any approximation order and any positive-definite self-adjoint spatial operators. Numerical experiments demonstrate the low condition numbers and fast convergence of the algorithm for both ideal and approximate preconditioners, and show the feasibility of the high-order solution of large problems.

Keywords: discontinuous Galerkin; time discretizations; parabolic PDE; preconditioning; conjugate gradient algorithm.

1. Introduction

The discontinuous Galerkin (DG) time-stepping method is a single-step implicit scheme defined by a variational temporal discretization of parabolic evolution equations that generalizes the backward Euler method to higher-order approximations (Delfour *et al.*, 1981; Eriksson *et al.*, 1985; Jamet, 1978; Hulme, 1972). For an introduction to this time discretization scheme in the context of abstract parabolic problems, we refer the reader to (Thomée, 2006). In certain cases, it coincides with the Radau IIA Implicit Runge–Kutta (IRK) schemes and the subdiagonal Padé approximations to the exponential function (Axelsson, 1969; Hairer & Wanner, 2010; Makridakis & Nochetto, 2006). It can be coupled with standard spatial discretization schemes, such as finite difference, finite element or spectral methods; in particular, when coupled to a spatial finite element method (FEM), it leads to a tensor-product space-time FEM.

The DG time-stepping method features many advantages that make it an attractive choice for solving parabolic problems. First, it permits arbitrarily high-order approximation to the solution, along with superconvergence at the time-step nodes (Akrivis & Makridakis, 2004; Chrysafinos & Walkington, 2006; Makridakis & Babuška, 1997), see also (Schötzau & Schwab, 2000) for optimal-order a priori error estimates in natural norms with explicit dependence on the polynomial degree. Unlike linear multistep schemes, it is thus not constrained by the Dahlquist barrier theorem and it does not require an auxiliary scheme to compute the first few solution values. Furthermore, the DG time-stepping method allows fully variable time-step sizes, approximation orders, and even spatial mesh refinement/coarsening between time-steps; it is thus well-suited for adaptive algorithms driven by rigorous a posteriori error estimates (Akrivis *et al.*, 2009; Eriksson & Johnson, 1991, 1995; Makridakis & Nochetto, 2006; Schötzau

†Corresponding author. Email:iain.smears@inria.fr

& Wihler, 2010). The DG time-stepping method also permits the temporal version of hp-refinement, where one varies both the time-step size as well as the approximation order on each time-step, thereby yielding exponential convergence rates for a broad class of solutions with singularities induced by the initial datum or by the source term (Schötzau & Schwab, 2000). In these applications, it is common to use high temporal polynomial degrees in order to match the high-order spatial approximation, see for instance the experiments in (Schötzau & Schwab, 2000; von Petersdorff & Schwab, 2004; Schötzau & Wihler, 2010; Werder *et al.*, 2001).

In applying the DG time-stepping method in practice, we are faced with the challenge of solving a large nonsymmetric linear system at each time-step. In this work, we are interested in developing preconditioned iterative methods for solving these linear systems. We focus here on the DG time-stepping method in the context of an abstract semi-discrete evolution problem of the form

$$M U'(t) + A U(t) = f(t) \quad \text{in } (0,T), \tag{1.1}$$

where the solution $U\colon [0,T] \to \mathsf{V}$, with V a finite dimensional space, where M and A are symmetric positive definite matrices. Semi-discrete problems of the form (1.1) arise from the application of a wide range of spatial discretizations, including conforming and nonconfirming finite elements, finite differences, and spectral methods, to a broad class of parabolic problems, such as general self-adjoint second-order and fourth-order parabolic partial differential equations (PDE). We emphasize that our results are valid for general symmetric positive definite matrices M and A, although the reader may consider the heat equation as a concrete example, with M and A respectively representing the mass and stiffness matrices for a suitable approximation space V. The DG time-stepping method applied to the evolution problem (1.1) leads to a sequence of linear systems of the general block form

$$\begin{bmatrix} b_{00} M + c_{00}\, \tau A & \cdots & b_{0p} M + c_{0p}\, \tau A \\ \vdots & \ddots & \vdots \\ b_{p0} M + c_{p0}\, \tau A & \cdots & b_{pp} M + c_{pp}\, \tau A \end{bmatrix} \begin{bmatrix} \mathsf{u}_0 \\ \vdots \\ \mathsf{u}_p \end{bmatrix} = \begin{bmatrix} \mathsf{f}_0 \\ \vdots \\ \mathsf{f}_p \end{bmatrix}, \tag{1.2}$$

where τ denotes the time-step size, the polynomial degree p defines the approximation order of the scheme, the solution coefficients $\mathsf{u}_k \in \mathsf{V}$ for $k = 0, \ldots, p$, and, after mapping the time-step interval to the reference interval $(-1,1)$, we have

$$b_{jk} \coloneqq \int_{-1}^{1} \phi_k' \, \phi_j \, \mathrm{d}s + \phi_k(-1)\phi_j(-1), \quad c_{jk} \coloneqq \frac{1}{2} \int_{-1}^{1} \phi_k \, \phi_j \, \mathrm{d}s, \tag{1.3}$$

where $\{\phi_k\}_{k=0}^{p}$ is a chosen basis of $\mathscr{P}_p$ the space of real-valued polynomials of degree at most p. For the case $p = 0$, the DG method reduces to the backward Euler method and the system is therefore symmetric. However, for $p \geqslant 1$, the system matrix $\mathbf{B}$ of (1.2) is nonsymmetric with dimension $\dim \mathsf{V} \times (p+1)$, which is considerably larger than for linear multistep methods; even for moderate sizes of A and M, standard direct solution algorithms can be prohibitively expensive. Unlike the block triangular linear systems obtained from Diagonally IRK (DIRK) and Singly Diagonally IRK (SDIRK) schemes (see (Hairer & Wanner, 2010) and the references therein), the system (1.2) does not immediately reveal any simple structure offering a straightforward solution algorithm. See also (Hairer & Wanner, 2010, p. 129–130) for a discussion of SIRK methods.

Since the DG time-stepping method is connected to the Radau IIA IRK scheme, it is interesting to relate our approach to the literature on solving the systems of IRK schemes. In particular, one of the earliest approaches (Butcher, 1976; Bickart, 1977) for solving the linear systems of general IRK

schemes, such as (1.2), is based on transforming the system matrix to a block-diagonal matrix with blocks of the form $M + \frac{\tau}{\mu_i}A$, where the $\{\mu_i\}_{i=0}^{p}$ denote here the generalized eigenvalues of (b_{jk}) with respect to (c_{jk}), and where the transformation is given by the corresponding eigenvectors. This leads to $p+1$ independent smaller linear systems that can be solved in parallel. It turns out that for the DG time-stepping method, the generalized eigenvalues μ_i are complex numbers related to the roots of the denominator of a rational Padé approximation to the exponential function. Therefore, the resulting transformed system is complex-valued and non-Hermitian despite M and A being symmetric. In addition to the increased computational cost of complex arithmetic, this approach has an important shortcoming in terms of robustness and numerical stability for high-order approximations, due to ill-conditioning of the eigenvector transformations, as pointed out in (von Petersdorff & Schwab, 2004, Remark 5.4). This issue can be avoided by employing alternative factorizations, at the expense of the block-diagonal structure of the transformed problem: Schötzau & Schwab (2000) propose a factorization based on the Schur decomposition theorem, leading to a block-triangular complex transformed problem; the solution is then obtained by solving $p+1$ complex non-Hermitian systems in sequence.

Preconditioned iterative methods offer an alternative approach to decoupling the system by complex transformations. In this direction, Richter *et al.* (2013) propose a linear iterative fixed point scheme based on an approximation of the block LU factorization of $\mathbf{B}$, and they analyse the contraction rates of their method for $p \leqslant 3$. An alternative approach is to apply directly a preconditioned Krylov subspace method to (1.2), as proposed in Mardal *et al.* (2007) propose a block-diagonal preconditioner for IRK schemes to be used with GMRES. They show that the preconditioner is robust with respect to the time-step size τ, but their experiments show that it is not robust with respect to p. Weller & Basting (2015) develop a preconditioner specifically for the DG time-stepping method for $p = 1$ (as well as for the related continuous Galerkin method of same size) based on an approximate Schur complement preconditioner for one of the unknown coefficients u_j in (1.2). It is thus apparent from these references that finding preconditioning strategies for the DG time-stepping method that are robust with respect to the polynomial degree p has been a challenging open problem.

In this work, we propose a robust and efficient preconditioned iterative method for the DG time-stepping method. Instead of focusing exclusively on the block structure of (1.2), our approach draws upon the inf-sup analysis of the method and the underlying continuous analysis of parabolic PDEs, and we take advantage of the variational structure of the DG time-stepping method in an essential way. First, in section 3.1, we construct and apply a left-preconditioner $\mathbf{P}^\top$ to the linear system $\mathbf{Bu} = \mathbf{f}$ given by (1.2), resulting in a preconditioned system $\mathbf{Lu} = \mathbf{g}$ with the key benefit that the transformed matrix $\mathbf{L} \coloneqq \mathbf{P}^\top\mathbf{B}$ is symmetric positive definite. We immediately point out that $\mathbf{P}^\top \neq \mathbf{B}^\top$, i.e. we are not forming the normal equations of the system. Instead, our construction of $\mathbf{P}^\top$ is motivated by parabolic inf-sup theory, and we show that the matrix $\mathbf{L}$ represents the discrete version of the natural parabolic energy norm of the underlying evolution equation: for example, in the context of second-order parabolic PDEs, the matrix $\mathbf{L}$ is a discrete Gram matrix for the natural solution space $H^1(H^{-1}) \cap L^2(H^1)$; we refer the reader to (Wloka, 1987) for an introduction to the continuous analysis of parabolic problems. The transformed symmetric positive definite system can therefore be solved by the preconditioned conjugate gradient (PCG) algorithm (Hestenes & Stiefel, 1952; Málek & Strakoš, 2015; Wathen, 2015), which, in our case, minimizes the error in the physically relevant norm over a Krylov subspace. In order to obtain the fast and robust convergence of the PCG algorithm, in section 3.2, we construct a spectrally equivalent preconditioner $\mathbf{H}$ for $\mathbf{L}$, such that the condition number of the preconditioned system satisfies

$$\kappa(\mathbf{H}^{-1}\mathbf{L}) \leqslant 4, \tag{1.4}$$

independently of all parameters τ, p, M and A. Therefore, the preconditioner is fully *robust* with respect

to all problem and discretization parameters, including the polynomial degree. Furthermore, the preconditioners are *efficient*, firstly in the sense of guaranteeing the fast convergence of the PCG algorithm in the physically relevant norm, and secondly in the sense of computational cost, for the following reasons. In section 4 we show that the preconditioners are well-suited for parallelization over the blocks and involve only simpler matrices of the form A and $M+\mu A$ with real positive μ for which efficient solvers are often available. Furthermore, we show experimentally in section 5 that the ideal preconditioners can be approximated in practice by cheap spectrally equivalent approximations, such as a small number of multigrid V-cycles. We refer the reader to (Wathen, 2015, p. 367) and (Hiptmair, 2006, p. 705) for further discussions of the notion of efficiency of preconditioners.

This paper is organized as follows: after introducing in detail the DG time-stepping method in section 2, we present the preconditioning strategy in section 3, where we construct the preconditioners $\mathbf{P}^\top$ and $\mathbf{H}$, and where we establish the condition number bound (1.4). Section 4 considers the efficient implementation of the method, and section 5 presents the results of numerical experiments testing the robustness and efficiency of the preconditioners.

2. Preliminaries

In this section, we introduce in detail the DG time-stepping method in the context of self-adjoint semi-discrete dissipative evolution equations. We also introduce two key ingredients in our approach. The first ingredient is the well-known temporal reconstruction operator commonly used in a posteriori analysis (Makridakis & Nochetto, 2006), which is the subject of section 2.3. The second ingredient is a spectral equivalence result for preconditioners from (Pearson & Wathen, 2012), which we present in section 2.4.

2.1 *Approximation space*

Let V denote a finite dimensional real vector space, equipped with a given basis $\{\mathsf{e}_i\}_{i=1}^{\dim\mathsf{V}}$. Let V be equipped with two inner products $(\cdot,\cdot)_M$ and $(\cdot,\cdot)_A$, and let M and A be their matrix representations, given by $M_{ij} \coloneqq (\mathsf{e}_i,\mathsf{e}_j)_M$ and $A_{ij} \coloneqq (\mathsf{e}_i,\mathsf{e}_j)_A$ for all $i,\ j=1,\dots,\dim\mathsf{V}$. The inner products $(\cdot,\cdot)_M$ and $(\cdot,\cdot)_A$ induce the norms $\|\cdot\|_M$ and $\|\cdot\|_A$ on V. Let $\mathscr{P}_p$ denote the space of real-valued polynomials of degree at most p, and let $\mathscr{V}_p$ be the space of V-valued polynomials of a single real variable with degree at most p. For example, if $\{\phi_j\}_{j=0}^p$ is a basis of $\mathscr{P}_p$, then every $v\in\mathscr{V}_p$ is of the form

$$v\colon s\mapsto v(s)=\sum_{j=0}^{p}\mathsf{v}_j\,\phi_j(s),$$

with coefficients $\mathsf{v}_j\in\mathsf{V}$ for each $j=0,\dots,p$. We gather these coefficients in the vector $\mathbf{v}=(\mathsf{v}_0,\dots,\mathsf{v}_p)\in\mathsf{V}^{p+1}$. It follows that $\dim\mathscr{V}_p$, the dimension of the space $\mathscr{V}_p$, is equal to $(p+1)\times\dim\mathsf{V}$. An equivalent point of view is to consider $\mathscr{V}_p$ as the tensor-product space derived from $\mathscr{P}_p$ and V.

REMARK 2.1 In this setting, it is natural to view functions in $\mathscr{V}_p$ as mappings from time into V, and as a slight abuse of standard terminology, we will say that $\{\phi_j\}_{j=0}^p$ forms a basis of $\mathscr{V}_p$. Of course, this must be interpreted as the lengthier statement that $\{\mathsf{e}_i\phi_j\}_{i=1,\dots,\dim\mathsf{V}}^{j=0,\dots,p}$ forms a basis of $\mathscr{V}_p$, where $\{\mathsf{e}_i\}_{i=1}^{\dim\mathsf{V}}$ is a basis of V.

If L is a linear operator between V and either itself or its dual V^*, then we can extend L to $\mathscr{V}_p$ by

applying it coefficient-wise:

$$Lv(s) \coloneqq \sum_{j=0}^{p} (L\mathsf{v}_j)\,\phi_j(s) \quad \forall v = \sum_{j=0}^{p} \mathsf{v}_j \phi_j \in \mathscr{V}_p. \tag{2.1}$$

The inner products $(\cdot,\cdot)_M$ and $(\cdot,\cdot)_A$ also extend to $\mathscr{V}_p$ in the natural way. Likewise, if $\pi\colon \mathscr{P}_p \to \mathscr{P}_p$ is a linear operator then we define $\pi v \coloneqq \sum_{j=0}^{p} \mathsf{v}_j\,(\pi\phi_j)$. For instance, we define the time derivative v' of $v \in \mathscr{V}_p$ by $v' \coloneqq \sum_{j=0}^{p} \mathsf{v}_j\,\phi_j'$.

Let $(\cdot,\cdot)_{L^2}$ and $\|\cdot\|_{L^2}$ denote respectively the L^2-inner product and L^2-norm over the interval $(-1,1)$. Let $\{L_k\}_{k\geqslant 0}$ denote the set of Legendre polynomials, as defined for instance in (Gradshteyn & Ryzhik, 2015, Sec. 8.9). The Legendre polynomials are orthogonal in the L^2-inner product: for all k, $j \geqslant 0$,

$$(L_k, L_j)_{L^2} = \frac{2}{2k+1}\,\delta_{kj}. \tag{2.2}$$

Furthermore, $L_k(1) = 1$ and $L_k(-1) = (-1)^k$ for all $k \geqslant 0$.

2.2 *The DG time-stepping method*

After mapping a given current time-step interval to the reference interval $(-1,1)$, the DG time-stepping method leads to the discrete problem of finding $u \in \mathscr{V}_p$ such that

$$\mathscr{B}(u,v) = \mathscr{F}(v) \qquad \forall v \in \mathscr{V}_p, \tag{2.3}$$

where $\mathscr{F}$ is a bounded linear functional on $\mathscr{V}_p$, and the bilinear form $\mathscr{B}\colon \mathscr{V}_p \times \mathscr{V}_p \to \mathbb{R}$ is defined by

$$\mathscr{B}(u,v) \coloneqq \int_{-1}^{1} \left(u', v\right)_M \mathrm{d}s + \left(u(-1), v(-1)\right)_M + \frac{\tau}{2}\int_{-1}^{1} (u,v)_A\,\mathrm{d}s, \tag{2.4}$$

where τ denotes the time-step size and p denotes the polynomial degree of the approximation. In practice, the time-step size, the polynomial degree, or even the matrices M and A may vary between time-steps, although on any given time-step the linear system has the general form of (2.3). Given a basis $\{\phi_j\}_{j=0}^{p}$ for $\mathscr{V}_p$, the problem (2.3) can be represented by a linear system

$$\mathbf{B}\mathbf{u} = \mathbf{f}, \tag{2.5}$$

where $\mathbf{B}$ is a $(p+1)\times(p+1)$ block matrix, where $\mathbf{u} \in \mathsf{V}^{p+1}$ is the vector of coefficients of the expansion of u, and where $\mathbf{f} = (\mathsf{f}_0, \cdots, \mathsf{f}_p)$ with each $\mathsf{f}_j \in \mathsf{V}^*$, $j = 0,\ldots,p$, being the restriction of $\mathscr{F}$ to $\operatorname{span}\phi_j \otimes \mathsf{V}$. Therefore, the system (2.5) has the block structure shown in (1.2).

REMARK 2.2 In order to motivate our approach to preconditioning, it will be helpful to bear in mind the following point concerning the structure of the bilinear form $\mathscr{B}$. It is well-known that the bilinear form $\mathscr{B}$ enjoys the following coercivity property

$$\mathscr{B}(v,v) = \frac{1}{2}\|v(1)\|_M^2 + \frac{1}{2}\|v(-1)\|_M^2 + \frac{\tau}{2}\int_{-1}^{1} \|v\|_A^2\,\mathrm{d}s \quad \forall v \in \mathscr{V}_p, \tag{2.6}$$

which enables us to deduce the well-posedness of (2.3). Unfortunately, the norm defined by the right-hand side of (2.6) does not include the time derivative, and thus coercivity and boundedness of $\mathscr{B}$ in

this norm can only be obtained from an inverse inequality for the finite dimensional space $\mathscr{V}_p$, at the expense of introducing constants that are not robust with respect to τ and p. This point suggests that norm preconditioners for $\mathbf{B}$ that are based on the right-hand side of (2.6) are unlikely to be robust with respect to the discretization parameters.

As we shall see below, our approach is based on the inf-sup stability of the bilinear form $\mathscr{B}$, which provides a sharper analysis of the structure of the problem than the coercivity result of (2.6). Some of our main tools are the reconstruction operator defined in section 2.3, and a corresponding suitable negative norm along with a key spectral equivalence result, which we recall in section 2.4.

2.3 *Reconstruction operator*

We introduce the reconstruction operator $\mathscr{I}: \mathscr{P}_p \to \mathscr{P}_{p+1}$, defined by

$$\mathscr{I}v := v - v(-1)\frac{(-1)^p\,(L_p - L_{p+1})}{2} \quad \forall v \in \mathscr{P}_p. \tag{2.7}$$

As noted above, $\mathscr{I}$ naturally extends to an operator from $\mathscr{V}_p$ to $\mathscr{V}_{p+1}$. As explained in Remark 2.3 below, $\mathscr{I}$ is the reconstruction operator commonly used in a posteriori error analysis (Makridakis & Nochetto, 2006). The key benefit of the reconstruction operator $\mathscr{I}$ for our purposes is that we may express the bilinear form $\mathscr{B}$ in the following equivalent form

$$\mathscr{B}(u,v) = \int_{-1}^{1} \big((\mathscr{I}u)', v\big)_M + \frac{\tau}{2}\,(u,v)_A \,\mathrm{d}s \quad \forall u, v \in \mathscr{V}_p. \tag{2.8}$$

We emphasize here that $(\mathscr{I}u)' \in \mathscr{V}_p$ since $\mathscr{I}u \in \mathscr{V}_{p+1}$ for $u \in \mathscr{V}_p$. To see how (2.8) is obtained from (2.7), we note that the properties of the Legendre polynomials imply that, for any $v \in \mathscr{V}_p$,

$$\mathscr{I}v(1) = v(1), \quad \mathscr{I}v(-1) = 0, \qquad \int_{-1}^{1} (\mathscr{I}v, w)_M \,\mathrm{d}s = \int_{-1}^{1} (v, w)_M \,\mathrm{d}s \quad \forall w \in \mathscr{V}_{p-1}. \tag{2.9}$$

Substituting w for w', which belongs to $\mathscr{V}_{p-1}$ whenever $w \in \mathscr{V}_p$, in (2.9) and using integration by parts shows that

$$\int_{-1}^{1} \big((\mathscr{I}v)', w\big)_M \,\mathrm{d}s = \int_{-1}^{1} (v', w)_M \,\mathrm{d}s + (v(-1), w(-1))_M \quad \forall w \in \mathscr{V}_p. \tag{2.10}$$

The equivalent form of $\mathscr{B}$ given in (2.8) then follows from (2.10).

REMARK 2.3 The operator $\mathscr{I}$ is the reconstruction operator commonly used in the a posteriori error analysis of the DG time-stepping method (Makridakis & Nochetto, 2006). This operator is often defined by the interpolation conditions $\mathscr{I}v(s_k) = v(s_k)$ for all $k = 1, \ldots, p+1$, where $-1 < s_k \leqslant s_{p+1} = 1$ are the Gauss–Radau quadrature points, in addition to a further condition, chosen here as $\mathscr{I}v(-1) = 0$. It turns out that the definition given above in (2.7) and this interpolatory definition are equivalent, since the Gauss–Radau points are the roots of the polynomial $(1-s)P_p^{(1,0)} = L_p - L_{p+1}$, see (Gautschi, 1997) and (Gradshteyn & Ryzhik, 2015, eq. 8.961.5). We note here that a straightforward consequence of the above properties is that, for any $v \in \mathscr{P}_p$, we have $v \equiv 0$ in $(-1,1)$ if and only if $(\mathscr{I}v)' \equiv 0$ in $(-1,1)$. We also note that we will not need the Gauss–Radau points for the implementation of the preconditioners in this work.

2.4 *Negative norms and a result of Pearson and Wathen*

In addition to the norms $\|\cdot\|_M$ and $\|\cdot\|_A$, we will also use the negative norm $\|\cdot\|_{MA^{-1}M}$ defined by

$$\|\mathsf{v}\|_{MA^{-1}M} \coloneqq \sup_{\mathsf{w}\in\mathsf{V}\setminus\{0\}} \frac{(\mathsf{v},\mathsf{w})_M}{\|\mathsf{w}\|_A} \quad \forall \mathsf{v}\in\mathsf{V}. \tag{2.11}$$

The negative norm $\|\cdot\|_{MA^{-1}M}$ can be equivalently characterized by the identity

$$\|\mathsf{v}\|^2_{MA^{-1}M} = \mathsf{v}^\top MA^{-1}M\mathsf{v} \quad \forall \mathsf{v}\in\mathsf{V}, \tag{2.12}$$

where v is identified with its vector representation in the basis $\{\mathsf{e}_i\}_{i=1}^{\dim\mathsf{V}}$. Indeed, (2.12) follows from the upper bound $\|\mathsf{v}\|^2_{MA^{-1}M} \geqslant \mathsf{v}^\top MA^{-1}M\mathsf{v}$, which is obtained by choosing $\mathsf{w} = A^{-1}M\mathsf{v}$ in (2.11), and from the lower bound $\|\mathsf{v}\|^2_{MA^{-1}M} \leqslant \mathsf{v}^\top MA^{-1}M\mathsf{v}$, which is obtained by applying the Cauchy–Schwarz inequality as follows: $\mathsf{w}^\top M\mathsf{v} = \mathsf{w}^\top A\left(A^{-1}M\mathsf{v}\right) \leqslant \sqrt{\mathsf{w}^\top A\,\mathsf{w}}\,\sqrt{\mathsf{v}^\top MA^{-1}M\mathsf{v}}$, where we have simplified $(A^{-1}M\mathsf{v})^\top A(A^{-1}M\mathsf{v}) = \mathsf{v}^\top MA^{-1}M\mathsf{v}$. The identity (2.12) thus shows that the norm $\|\cdot\|_{MA^{-1}M}$ is in fact induced by an inner product $(\cdot,\cdot)_{MA^{-1}M}$ represented by the matrix $MA^{-1}M$.

The following result due to J. W. Pearson and A. J. Wathen (Pearson & Wathen, 2012) will play a key part in the construction of our preconditioners.

LEMMA 2.1 Let A and M be arbitrary symmetric positive definite matrices and let $\mu \geqslant 0$ be a nonnegative real number. Then we have

$$\frac{1}{2} \leqslant \frac{\mathsf{v}^\top\left(MA^{-1}M+\mu A\right)\mathsf{v}}{\mathsf{v}^\top\left(M+\sqrt{\mu}A\right)A^{-1}\left(M+\sqrt{\mu}A\right)\mathsf{v}} \leqslant 1 \quad \forall \mathsf{v}\in\mathsf{V}\setminus\{0\}. \tag{2.13}$$

Proof. For the original proof of this result, see (Pearson & Wathen, 2012, Thm 4). We provide here an alternative proof which highlights the negative norm structure of the matrix $MA^{-1}M$, as given in (2.11). First, the upper bound of (2.13) is immediate from

$$\|\mathsf{v}\|^2_{MA^{-1}M} + \mu\|\mathsf{v}\|^2_A \leqslant \|\mathsf{v}\|^2_{MA^{-1}M} + 2\sqrt{\mu}\|\mathsf{v}\|^2_M + \mu\|\mathsf{v}\|^2_A.$$

Now, for any $\mathsf{v}\in\mathsf{V}$, we have $\|\mathsf{v}\|^2_M \leqslant \|\mathsf{v}\|_{MA^{-1}M}\|\mathsf{v}\|_A$ by (2.11). It follows from the inequality $ab \leqslant \frac{1}{2}a^2 + \frac{1}{2}b^2$ for any $a, b\in\mathbb{R}$ that

$$\begin{aligned}\mathsf{v}^\top\left(M+\sqrt{\mu}A\right)A^{-1}\left(M+\sqrt{\mu}A\right)\mathsf{v} &= \|\mathsf{v}\|^2_{MA^{-1}M} + 2\sqrt{\mu}\|\mathsf{v}\|^2_M + \mu\|\mathsf{v}\|^2_A \\ &\leqslant 2\|\mathsf{v}\|^2_{MA^{-1}M} + 2\mu\|\mathsf{v}\|^2_A = 2\,\mathsf{v}^\top\left(MA^{-1}M+\mu A\right)\mathsf{v}.\end{aligned} \tag{2.14}$$

Note that (2.14) is precisely the lower bound of (2.13). □

3. Preconditioners

3.1 *Left preconditioner*

In this section, we construct a left preconditioner that will transform the linear system (2.5) to a symmetric positive definite system that represents a discrete parabolic energy norm. Our left preconditioner is defined simply in terms of a substitution for the test function v appearing in the bilinear form $\mathscr{B}(u,v)$. We start by defining the operator $P\colon \mathscr{V}_p \to \mathscr{V}_p$ by

$$Pv \coloneqq A^{-1}M(\mathscr{I}v)' + \frac{\tau}{2}v, \tag{3.1}$$

where we recall the definition of the reconstruction operator $\mathscr{I}$ from (2.7) and its natural to extension to $\mathscr{V}_p$ as explained in section 2.1. The fact that $Pv \in \mathscr{V}_p$ for any $v \in \mathscr{V}_p$ implies that the solution u of (2.3) also solves

$$\mathscr{L}(u,v) = \mathscr{G}(v) \qquad \forall v \in \mathscr{V}_p, \tag{3.2}$$

where the bilinear form $\mathscr{L}$ and linear functional $\mathscr{G}$ are obtained by substituting the test function Pv in place of v:

$$\mathscr{L}(u,v) \coloneqq \mathscr{B}(u,Pv), \quad \mathscr{G}(v) \coloneqq \mathscr{F}(Pv) \quad \forall u, v \in \mathscr{V}_p. \tag{3.3}$$

We note that the operator P can be viewed as defining a left preconditioner for the linear system (2.5) that represents (2.3). Indeed, let $\mathbf{P}$ denote the matrix representation of the linear operator P in the basis $\{\phi_j\}_{j=0}^p$. Then, for any functions u and $v \in \mathscr{V}_p$, and their respective vector representations $\mathbf{u}$ and $\mathbf{v} \in \mathsf{V}^{p+1}$, we have $\mathscr{B}(u,Pv) = \mathbf{v}^\top \mathbf{P}^\top \mathbf{B}\mathbf{u}$ and $\mathscr{F}(Pv) = \mathbf{v}^\top \mathbf{P}^\top \mathbf{f}$. Therefore, the linear system (3.2) is equivalent to

$$\mathbf{L}\mathbf{u} = \mathbf{g}, \tag{3.4}$$

with $\mathbf{L} = \mathbf{P}^\top \mathbf{B}$ and $\mathbf{g} = \mathbf{P}^\top \mathbf{f}$ denoting respectively the left-preconditioned matrix and right-hand side.

As the following theorem shows, $\mathscr{L}$ represents a discrete version of the natural energy norm for parabolic problems. Indeed, in applications to second-order parabolic PDEs, $\mathscr{L}$ is comparable to the inner product of $L^2(H^1) \cap H^1(H^{-1})$, which is the natural solution space of the continuous problem (Wloka, 1987).

THEOREM 3.1 Let the bilinear form $\mathscr{L}\colon \mathscr{V}_p \times \mathscr{V}_p \to \mathbb{R}$ and linear functional $\mathscr{G}\colon \mathscr{V}_p \to \mathbb{R}$ be defined by (3.3). Then, for any functions u and $v \in \mathscr{V}_p$, we have the identity

$$\begin{aligned} \mathscr{L}(u,v) = \int_{-1}^{1} \left((\mathscr{I}u)', (\mathscr{I}v)'\right)_{MA^{-1}M} \mathrm{d}s + \frac{\tau^2}{4} \int_{-1}^{1} (u,v)_A \,\mathrm{d}s \\ + \frac{\tau}{2} \left(u(1), v(1)\right)_M + \frac{\tau}{2} \left(u(-1), v(-1)\right)_M \end{aligned} \tag{3.5}$$

Therefore, $\mathscr{L}$ is symmetric and positive definite on $\mathscr{V}_p$, and for any symmetric positive definite matrices A and M, any $p \geqslant 0$ and any $\tau > 0$, we have

$$\mathscr{L}(v,v) \geqslant \|v\|_{\mathscr{D}}^2, \quad |\mathscr{L}(v,w)| \leqslant 2\|v\|_{\mathscr{D}} \|w\|_{\mathscr{D}} \qquad \forall v, w \in \mathscr{V}_p, \tag{3.6}$$

where $\|\cdot\|_{\mathscr{D}} \coloneqq \sqrt{\mathscr{D}(\cdot,\cdot)}$ is the norm induced by the auxiliary bilinear form $\mathscr{D}$ defined by:

$$\mathscr{D}(u,v) \coloneqq \int_{-1}^{1} \left((\mathscr{I}u)', (\mathscr{I}v)'\right)_{MA^{-1}M} \mathrm{d}s + \frac{\tau^2}{4} \int_{-1}^{1} (u,v)_A \mathrm{d}s. \tag{3.7}$$

Thus, the function $u \in \mathscr{V}_p$ solves (2.3) if and only if u solves (3.2).

Proof. First, it is straightforward to obtain the following identities which hold for any $u, v \in \mathscr{V}_p$:

$$\left((\mathscr{I}u)', A^{-1}M(\mathscr{I}v)'\right)_M = \left((\mathscr{I}u)', (\mathscr{I}v)'\right)_{MA^{-1}M}, \quad \left(u, A^{-1}M(\mathscr{I}v)'\right)_A = (u, (\mathscr{I}v)')_M. \tag{3.8}$$

Next, we use the identities in (3.8) to simplify the different terms in $\mathscr{L}(u,v) = \mathscr{B}(u,Pv)$, and eventually we obtain

$$\begin{aligned} \mathscr{L}(u,v) = \int_{-1}^{1} \left((\mathscr{I}u)', (\mathscr{I}v)'\right)_{MA^{-1}M} \mathrm{d}s + \frac{\tau^2}{4} \int_{-1}^{1} (u,v)_A \,\mathrm{d}s \\ + \frac{\tau}{2} \int_{-1}^{1} \left((\mathscr{I}u)', v\right)_M + \left(u, (\mathscr{I}v)'\right)_M \mathrm{d}s. \end{aligned} \tag{3.9}$$

By (2.10), we have

$$\int_{-1}^{1} \left((\mathscr{I}u)', v\right)_M + \left(u, (\mathscr{I}v)'\right)_M \mathrm{d}s = \int_{-1}^{1} \frac{\mathrm{d}}{\mathrm{d}s}(u,v)_M \,\mathrm{d}s + 2\,(u(-1), v(-1))_M$$
$$= (u(1), v(1))_M + (u(-1), v(-1))_M\,,$$

which, after substitution of the last terms in (3.9), implies the equivalent form for $\mathscr{L}$ given in (3.5). Next, to show (3.6), we note that first that the lower bound $\mathscr{L}(v,v) \geqslant \|v\|_{\mathscr{D}}^2$ is immediate, whereas the upper bound $|\mathscr{L}(v,w)| \leqslant 2\|v\|_{\mathscr{D}}\|w\|_{\mathscr{D}}$ follows from the application of the Cauchy–Schwarz inequality to (3.9). Finally, it follows that the problem (3.2) has a unique solution, momentarily denoted $\tilde{u} \in \mathscr{V}_p$. Moreover, it is well-known that (2.3) has a unique solution $u \in \mathscr{V}_p$. Since the definition of $\mathscr{L}$ and $\mathscr{G}$ in (3.3) shows that u is a solution of (3.2), we deduce from uniqueness that $\tilde{u} = u$. Therefore, the problems (2.3) and (3.2) are equivalent. □

REMARK 3.1 Theorem 3.1 can be viewed as part of the inf-sup analysis of the DG time-stepping method: defining the energy norm $\|\cdot\|_{\mathscr{L}} := \sqrt{\mathscr{L}(\cdot,\cdot)}$ and the norm $\|\cdot\|_{\mathscr{X}} := \sqrt{\int_{-1}^{1}\|\cdot\|_A^2 \mathrm{d}s}$, we have

$$\|u\|_{\mathscr{L}} = \sup_{v \in \mathscr{V}_p \setminus \{0\}} \frac{\mathscr{B}(u,v)}{\|v\|_{\mathscr{X}}}, \quad |\mathscr{B}(u,v)| \leqslant \|u\|_{\mathscr{L}}\|v\|_{\mathscr{X}} \qquad \forall\, u, v \in \mathscr{V}_p, \tag{3.10}$$

where the first equality is attained by choosing the optimal test function $v = Pu$, since $\|Pu\|_{\mathscr{X}} = \|u\|_{\mathscr{L}}$. This observation highlights the fact that the operator P represents the operator that gives the optimal test function in the inf-sup analysis of the DG time-stepping method. It is in this sense that our preconditioning strategy is directly motivated by the inf-sup theory of the DG time-stepping method.

3.2 *Spectrally equivalent norm preconditioner*

As shown by Theorem 3.1, the bilinear form $\mathscr{L}$ is symmetric and positive definite. Therefore, the linear system (3.4) can be solved iteratively by the preconditioned conjugate gradient (PCG) algorithm (Hestenes & Stiefel, 1952; Málek & Strakoš, 2015; Wathen, 2015). In this section, we construct a spectrally equivalent and easily applicable preconditioner for $\mathbf{L}$, thereby leading to the robust and fast convergence of the PCG algorithm. Recall that the bilinear form $\mathscr{L}$ is spectrally equivalent to the bilinear form $\mathscr{D}$ defined by (3.7). The first step in our construction of a preconditioner is to choose an advantageous temporal basis for $\mathscr{V}_p$ that block-diagonalizes the matrix $\mathbf{D}$ that represents $\mathscr{D}$. In a second step, we construct an easily applicable preconditioner, denoted by $\mathbf{H}$, under this basis, and we apply Lemma 2.1 to show robust spectral equivalence between $\mathbf{H}$ and $\mathbf{L}$.

3.2.1 *Definition of the basis.* It follows from Remark 2.3 that the symmetric bilinear form $(u,v) \to \int_{-1}^{1}(\mathscr{I}u)'(\mathscr{I}v)'\,\mathrm{d}s$ is positive definite and thus defines an inner-product on $\mathscr{P}_p$. Therefore, there exists a set of linearly independent polynomial eigenfunctions $\{\varphi_j\}_{j=0}^{p} \subset \mathscr{P}_p$ and corresponding real positive eigenvalues $\{\lambda_j\}_{j=0}^{p} \subset \mathbb{R}_{>0}$ such that

$$\lambda_j \int_{-1}^{1} (\mathscr{I}\varphi_j)'\,(\mathscr{I}v)'\,\mathrm{d}s = \int_{-1}^{1} \varphi_j\, v\,\mathrm{d}s \qquad \forall\, v \in \mathscr{P}_p,\ j = 0, \ldots, p. \tag{3.11}$$

We note that the eigenvalues and eigenfunctions generally depend on p, and are not necessarily hierarchical. The eigenfunctions are chosen to be orthonormalized:

$$\int_{-1}^{1} (\mathscr{I}\varphi_j)'\,(\mathscr{I}\varphi_k)'\,\mathrm{d}s = \delta_{jk} \quad \forall 0 \leqslant j,k \leqslant p. \tag{3.12}$$

We provide a practical method for computing this basis along with the corresponding eigenvalues in section 4.1. The following result characterizes the distribution of the eigenvalues.

THEOREM 3.2 Let $p \geqslant 0$ be a nonnegative integer, and let $\lambda_0 \geqslant \cdots \geqslant \lambda_p$ be the eigenvalues of the generalized eigenvalue problem (3.11). Then, there exists a positive constant C, independent of j and p, such that

$$\lambda_j \leqslant C(j+1)^{-2} \quad \forall\, j = 0,\ldots,p,\ \forall\, p \geqslant 0. \tag{3.13}$$

Moreover, there exists a constant c, independent of p, such that

$$\lambda_0 \geqslant \cdots \geqslant \lambda_p \geqslant c\,(p+1)^{-4}. \tag{3.14}$$

We leave the proof of Theorem 3.2 to Appendix A since it is based on some results presented in the later section 4.1. Figure 2 shows that the orders of the bounds of Theorem 3.2 are sharp. The eigenfunctions for $p = 4$ are shown in Figure 1.

3.2.2 *Construction of the norm preconditioner.* It is advantageous to use the basis $\{\varphi_j\}_{j=0}^{p}$ in computations since in this basis, the dominant terms in the bilinear form $\mathscr{L}$, namely those belonging to the bilinear form $\mathscr{D}$, are represented by a block-diagonal matrix. Indeed, let $\mathbf{D}$ denote the matrix representation of the bilinear form $\mathscr{D}$, defined in (3.7), under the basis $\{\varphi_j\}_{j=0}^{p}$ of $\mathscr{V}_p$, where we recall the terminology from Remark 2.1. Then, in this basis, the matrix $\mathbf{D}$ has a simple block-diagonal structure:

$$\mathbf{D} = \operatorname{diag}\left\{ MA^{-1}M + \frac{\tau^2\lambda_j}{4}A \right\}_{j=0}^{p}. \tag{3.15}$$

In other words, $\mathbf{D}$ is block-diagonal, with blocks $MA^{-1}M + \lambda_j\frac{\tau^2}{4}A$ for $j = 0,\ldots,p$. Keeping in mind Lemma 2.1, it is then natural to define the norm preconditioner $\mathbf{H}$ by

$$\mathbf{H} \coloneqq \operatorname{diag}\left\{ \left(M + \frac{\tau\sqrt{\lambda_j}}{2}A\right) A^{-1} \left(M + \frac{\tau\sqrt{\lambda_j}}{2}A\right) \right\}_{j=0}^{p}. \tag{3.16}$$

The matrix $\mathbf{H}$ is symmetric positive definite, and its inverse is trivially given by

$$\mathbf{H}^{-1} = \operatorname{diag}\left\{ \left(M + \frac{\tau\sqrt{\lambda_j}}{2}A\right)^{-1} A \left(M + \frac{\tau\sqrt{\lambda_j}}{2}A\right)^{-1} \right\}_{j=0}^{p}.$$

We propose to use $\mathbf{H}$ as a preconditioner for the PCG algorithm applied to (3.4). Each PCG iteration requires the application of $\mathbf{H}^{-1}$, which comprises two applications of a solver for a weighted backward Euler step and one multiplication by A per block.

We now give the central result of this work, which shows that $\mathbf{L}$ and $\mathbf{H}$ are spectrally equivalent with fully robust bounds.

THEOREM 3.3 Let $p \geqslant 0$, and let $\{\varphi_j\}_{j=0}^p \subset \mathscr{P}_p$ and $\{\lambda_j\}_{j=0}^p \subset \mathbb{R}_{>0}$ be the eigenfunctions and eigenvalues of (3.11). Let $\mathbf{L}$ be the matrix representation of $\mathscr{L}$ in the basis $\{\varphi_j\}_{j=0}^p$. Then, for any symmetric positive definite matrices A and M, any $\tau > 0$ and any $p \geqslant 0$, we have

$$\frac{1}{2} \leqslant \frac{\mathbf{v}^\top \mathbf{L}\mathbf{v}}{\mathbf{v}^\top \mathbf{H}\mathbf{v}} \leqslant 2 \quad \forall \mathbf{v} \in \mathsf{V}^{p+1} \setminus \{0\}. \tag{3.17}$$

Proof. Theorem 3.1, in particular (3.6), shows that

$$1 \leqslant \frac{\mathbf{v}^\top \mathbf{L}\mathbf{v}}{\mathbf{v}^\top \mathbf{D}\mathbf{v}} \leqslant 2 \quad \forall \mathbf{v} \in \mathsf{V}^{p+1} \setminus \{0\}. \tag{3.18}$$

Lemma 2.1, applied block-wise with $\mu = \lambda_j \tau^2/4$, implies that

$$\frac{1}{2} \leqslant \frac{\mathbf{v}^\top \mathbf{D}\mathbf{v}}{\mathbf{v}^\top \mathbf{H}\mathbf{v}} \leqslant 1 \quad \forall \mathbf{v} \in \mathsf{V}^{p+1} \setminus \{0\}. \tag{3.19}$$

Therefore, (3.18) and (3.19) imply (3.17). □

Theorem 3.3 immediately implies that the condition number κ of the preconditioned system, defined as the ratio of extremal eigenvalues of $\mathbf{H}^{-1}\mathbf{L}$, satisfies

$$\kappa \leqslant 4. \tag{3.20}$$

Therefore, we may expect very fast convergence from the PCG algorithm for $\mathbf{L}\mathbf{u} = \mathbf{g}$ using the preconditioner $\mathbf{H}$. Indeed, for an initial guess $u_0 \in \mathscr{V}_p$, the iterates $\{u_k\}_{k\geqslant 0}$ of the PCG algorithm satisfy the well-known bound (Wathen, 2015)

$$\frac{\|u-u_k\|_{\mathscr{L}}}{\|u-u_0\|_{\mathscr{L}}} \leqslant 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \leqslant \frac{2}{3^k} \quad \forall k \geqslant 0, \tag{3.21}$$

where we recall the energy norm $\|\cdot\|_{\mathscr{L}} = \sqrt{\mathscr{L}(\cdot,\cdot)}$. The fact that the condition number $\kappa \leqslant 4$ shows that the left preconditioner $\mathbf{P}^\top$ and the norm preconditioner $\mathbf{H}$ are very effective at preconditioning the original system matrix $\mathbf{B}$, despite $\mathbf{B}$ being nonsymmetric and poorly conditioned. This is a key aspect of the efficiency of the proposed preconditioners. Furthermore, we highlight that (3.20) and (3.21) are valid for any time-step size τ, any polynomial degree p, and any symmetric positive definite matrices M and A. Thus the preconditioners are fully robust with respect to all discretization and problem parameters.

REMARK 3.2 The fact that the energy norm $\|\cdot\|_{\mathscr{L}}$ appears in the bound (3.21) is advantageous in practice. Since the endpoint value of the solution at $s = 1$ serves as initial datum for the next time-step, it is beneficial to be able to control the accuracy to which it is computed. For example, (3.5) shows that the energy norm controls the value at $s = 1$ in the M-norm, which is the natural norm for this quantity of interest. We note that the factor of τ appearing alongside $\|v(1)\|_M^2$ in the energy norm can be scaled out as it appears in both the numerator and denominator of (3.21). Therefore (3.21) is a guaranteed convergence rate for this quantity of interest in the physically relevant norm.

4. Implementation

4.1 *Computation of eigenfunctions and eigenvalues*

The algorithm requires the computation of the eigenfunctions $\{\varphi_j\}_{j=0}^p$ and eigenvalues $\{\lambda_j\}_{j=0}^p$ defined in (3.11), which involves solving a symmetric positive definite eigenvalue problem of dimension $p+1$.

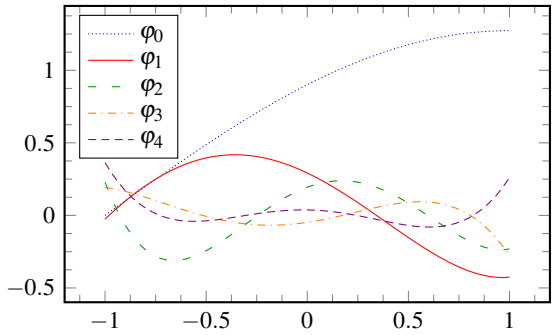

FIG. 1. Eigenfunctions $\{\varphi_j\}_{j=0}^{p}$ defined by (3.11) and (3.12), computed for $p = 4$ by the method of section 4.1 and ordered by decreasing eigenvalue $\lambda_j \geqslant \lambda_{j+1}$. Note that the j-th eigenfunction φ_j need not be of degree at most j. Furthermore, the set of eigenfunctions generally depends on p, although they are independent of A, M and τ.

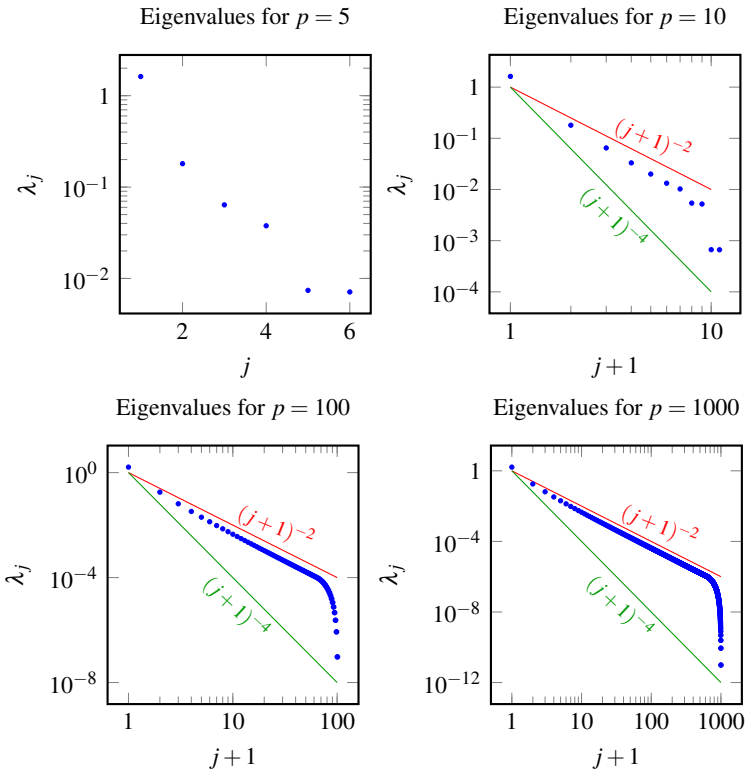

FIG. 2. Eigenvalues $\{\lambda_j\}_{j=0}^{p}$ computed by the method of section 4.1 and arranged in decreasing order, for $p = 5$, 10, 100 and 100. All plots are on a logarithmic scale, except for $p = 5$ which is given on a semilogarithmic scale. It appears that the bounds of Theorem 3.2 are sharp.

This poses little difficulty, as in practice p is usually small, especially in comparison to $\dim \mathsf{V}$. Since this step only depends on p and is fully independent of A, M and τ, it can be pre-computed to very high accuracy. We now show how to assemble this eigenvalue problem in a form where it can be solved numerically. Since the cases $p = 0$ and $p = 1$ can be easily computed by hand, we present a general method for $p \geqslant 2$. In the following, we will use the following identity for the Legendre polynomials $\{L_k\}_{k=0}^p$, see (Gradshteyn & Ryzhik, 2015, Sec. 8.914):

$$L_k = \frac{L'_{k+1} - L'_{k-1}}{2k+1} \quad \forall k \geqslant 0. \tag{4.1}$$

LEMMA 4.1 Let $p \geqslant 2$ be a nonnegative integer. Define the polynomials

$$\begin{aligned} \psi_0 &:= \frac{1}{\sqrt{2}}(L_1 + L_0) \quad \psi_p := \frac{L_p - L_{p-1}}{\sqrt{4p+2}}, \\ \psi_k &:= \frac{L_{k+1} - L_{k-1}}{\sqrt{4k+2}}, \quad k = 1, \dots, p-1. \end{aligned} \tag{4.2}$$

Then, there holds

$$(\mathscr{I}\psi_k)' = \sqrt{k+\tfrac{1}{2}}\, L_k \quad \forall k = 0, \dots, p. \tag{4.3}$$

Therefore, we have, for each $0 \leqslant k, j \leqslant p$,

$$\int_{-1}^{1} (\mathscr{I}\psi_k)'(\mathscr{I}\psi_j)' \,\mathrm{d}s = \delta_{kj}. \tag{4.4}$$

Proof. Consider the case $k = 0$: $\psi_0(-1) = 0$, and $L'_1 = L_0$. Therefore $\mathscr{I}\psi_0 = \psi_0$, and thus $(\mathscr{I}\psi_0)' = L_0/\sqrt{2}$, which is (4.3) for $k = 0$. Now consider the general case $k = 1, \dots, p-1$. Since $L_j(-1) = (-1)^j$ for all $j \geqslant 0$, we have $\psi_k(-1) = 0$ and thus $\mathscr{I}\psi_k = \psi_k$. Now, the identity (4.1) implies that

$$(\mathscr{I}\psi_k)' = \frac{2k+1}{\sqrt{4k+2}} L_k = \sqrt{k+\tfrac{1}{2}}\, L_k, \tag{4.5}$$

thus verifying (4.3) for $k = 1, \dots, p-1$. For the case $k = p$, we use the fact that $L_p(-1) - L_{p-1}(-1) = 2(-1)^p$ to compute

$$\mathscr{I}(L_p - L_{p-1}) = L_p - L_{p-1} - \frac{2(-1)^p(-1)^p}{2}(L_p - L_{p+1}) = L_{p+1} - L_{p-1}. \tag{4.6}$$

Therefore, similarly to (4.5), the identity (4.3) for $k = p$ follows from (4.1). □

Define the matrix $Z \in \mathbb{R}^{(p+1)\times(p+1)}$ such that $\psi_k = \sum_{j=0}^p Z_{kj} L_j$ as in (4.2). Define the diagonal matrix $D := \operatorname{diag}\{2/(2j+1)\}_{j=0}^p$, which represents the L^2-inner product in the Legendre polynomial basis. Define the matrix $T \in \mathbb{R}^{(p+1)\times(p+1)}$ by

$$T_{kj} := \int_{-1}^{1} \psi_k \psi_j \,\mathrm{d}s, \quad k, j = 0, \dots, p. \tag{4.7}$$

Note that $T = ZDZ^\top$. It follows from (4.4) that the eigenvalue problem (3.11) can be expressed as: find $V \in \mathbb{R}^{(p+1)\times(p+1)}$ and $\{\lambda_j\}_{j=0}^p$ such that

$$TV = V \operatorname{diag}\{\lambda_j\}_{j=0}^p, \quad VV^\top = V^\top V = \mathrm{Id}. \tag{4.8}$$

The eigenvalue problem (4.8) can thus be solved numerically by standard eigenvalue solvers. The eigenfunctions $\{\varphi_j\}_{j=0}^p$ can then be recovered as

$$\varphi_j(s) = \sum_{k=0}^{p} V_{kj}\,\psi_k(s) = \sum_{k=0}^{p} Q_{kj}\,L_k(s), \quad Q := Z^\top V. \tag{4.9}$$

4.2 *Implementation of preconditioners*

In order to compute the action of the left-preconditioner $\mathbf{P}^\top$, we use the matrix representation of the mapping $v \mapsto (\mathscr{I}v)'$ in the basis given by $\{\varphi_j\}_{j=0}^p$. Thus we need to find the matrix K such that $(\mathscr{I}\varphi_k)' = \sum_{j=0}^p K_{kj}\,\varphi_j$ for all $k = 0,\dots,p$. We show below that this is easily computed. Inverting (4.1) by induction yields

$$L_k' = \sum_{j=0}^{k-1} \left(1-(-1)^{k-j}\right)\left(j+\tfrac{1}{2}\right) L_j \quad \forall k \geqslant 0. \tag{4.10}$$

Therefore, we have

$$(\mathscr{I}L_k)' = \sum_{j=0}^{p} K^*_{kj} L_j \quad \forall k = 0,\dots,p, \tag{4.11}$$

where, after some calculation, it is found that the matrix $K^* \in \mathbb{R}^{(p+1)\times(p+1)}$ can be defined in terms of $\tilde{K} \in \mathbb{R}^{(p+1)\times(p+1)}$, $x \in \mathbb{R}^{p+1}$, and $y \in \mathbb{R}^{p+1}$ by

$$K^* := \tilde{K} - xy^\top, \qquad \tilde{K}_{kj} := \begin{cases} \left(1-(-1)^{k-j}\right)\left(j+\frac{1}{2}\right) & \text{if } j \leqslant k, \\ 0 & \text{otherwise,} \end{cases}$$
$$x_j := (-1)^j, \qquad y_k := (-1)^{1-k}\left(k+\tfrac{1}{2}\right).$$

LEMMA 4.2 Let $p \geqslant 2$ be an integer, and let $\{\varphi_j\}_{j=0}^p$ be the eigenfunctions defined by (4.9). Then, for each $k = 0,\dots,p$, there holds

$$(\mathscr{I}\varphi_k)' = \sum_{j=0}^{p} K_{kj}\,\varphi_j, \quad K := V^\top D^{-1/2} K^* D^{1/2} V, \tag{4.12}$$

where V is as in (4.8), the matrix K^* is as in (4.11), and where $D = \operatorname{diag}\{2/(2j+1)\}_{j=0}^p$.

Proof. The orthogonality of the matrix V in (4.8) implies that $\psi_k = \sum_{m=0}^p V_{km}\varphi_m$. Since the basis $\{\psi_k\}_{k=0}^p$ is orthonormal in the inner product of (4.4), there holds

$$\begin{aligned} L_j &= \sum_{r=0}^{p} \left(\int_{-1}^{1} (\mathscr{I}L_j)'(\mathscr{I}\psi_r)'\,\mathrm{d}s \right) \psi_r = \sum_{r=0}^{p}\sum_{m=0}^{p} \left(K^*_{jm}(r+\tfrac{1}{2})^{1/2} \int_{-1}^{1} L_m L_r\,\mathrm{d}s \right) \psi_r \\ &= \sum_{r=0}^{p} K^*_{jr}\left(r+\tfrac{1}{2}\right)^{-1/2} \psi_r = \sum_{r=0}^{p} (K^* D^{1/2})_{jr}\,\psi_r = \sum_{m=0}^{p} (K^* D^{1/2} V)_{jm}\,\varphi_m, \end{aligned}$$

where we have used (4.3) and (4.11) in the first line. Therefore, we compute

$$(\mathscr{I}\varphi_k)' = \sum_{j=0}^{p} V_{jk}(\mathscr{I}\psi_j)' = \sum_{j=0}^{p} V_{jk}(j+\tfrac{1}{2})^{1/2} L_j = \sum_{j=0}^{p}\sum_{m=0}^{p} V_{jk}(D^{-1/2}K^*D^{1/2}V)_{jm}\varphi_m,$$

which completes the proof. □

We now show how the matrix K is used in applying the preconditioner $\mathbf{P}^\top$. For $v = \sum_{k=0}^{p} \mathsf{v}_k \varphi_k$, there holds

$$Pv = \sum_{j=0}^{p} \left(A^{-1} M \mathsf{w}_j + \frac{\tau}{2} \mathsf{v}_j \right) \varphi_j, \quad \mathsf{w}_j := \sum_{k=0}^{p} K_{kj} \mathsf{v}_k,$$

$$\mathscr{F}(Pv) = \sum_{j=0}^{p} \mathsf{f}_j^\top \left(A^{-1} M \mathsf{w}_j + \frac{\tau}{2} \mathsf{v}_j \right) = \sum_{k=0}^{p} \left(M A^{-1} \left(\sum_{j=0}^{p} K_{kj} \mathsf{f}_j \right) + \frac{\tau}{2} \mathsf{f}_k \right)^\top \mathsf{v}_k.$$

Therefore, $\mathbf{g} := \mathbf{P}^\top \mathbf{f}$, where $\mathbf{g} = (\mathsf{g}_0, \ldots, \mathsf{g}_p)$, can be computed componentwise by

$$\mathsf{g}_k = M A^{-1} \left(\sum_{j=0}^{p} K_{kj} \mathsf{f}_j \right) + \frac{\tau}{2} \mathsf{f}_k, \quad k = 0, \ldots, p. \tag{4.13}$$

The action of $\mathbf{P}^\top$ requires the solution of $p+1$ independent systems with matrix A, which can be performed in parallel. If we ignore communication costs, then the cost of computing $\mathbf{g}$ is independent of the polynomial degree p on a parallel machine with sufficiently many computing nodes.

After application of the preconditioner $\mathbf{P}^\top$, the linear system has the form

$$\mathbf{L}\mathbf{u} = \mathbf{g},$$

where $\mathbf{L} = \mathbf{P}^\top \mathbf{B}$, which can be solved by the PCG algorithm with preconditioner $\mathbf{H}$, as suggested in section 3. The PCG algorithm requires the action of the matrices $\mathbf{L}$ and $\mathbf{H}^{-1}$ at each iteration. There are two ways to implement the action of $\mathbf{L}$, the first being the application of $\mathbf{B}$ followed by $\mathbf{P}^\top$. The downside of this approach is that it leads to greater communication costs on a distributed memory parallel machine where each node holds in memory only a few coefficients of $\mathbf{u}$: the matrices $\mathbf{B}$ and $\mathbf{P}^\top$ are generally block-dense, so all vector components must be exchanged between all computational nodes for both steps. The second approach is to use (3.5), which shows that $\mathbf{L}$ can be expressed as a diagonal matrix plus a "rank-two" term: in the basis $\{\varphi_j\}_{j=0}^{p}$, we have

$$\mathbf{L} = \operatorname{diag}\left\{ M A^{-1} M + \frac{\tau^2 \lambda_j}{4} A \right\}_{j=0}^{p} + \frac{\tau}{2} \mathbf{q}_1 \mathbf{q}_1^\top \otimes M + \frac{\tau}{2} \mathbf{q}_{-1} \mathbf{q}_{-1}^\top \otimes M, \tag{4.14}$$

where $\mathbf{q}_{\pm 1} := (q_0(\pm 1), \ldots, q_p(\pm 1))$ are the vectors of endpoint values of the φ_j. Therefore, we may compute $\mathbf{L}\mathbf{u}$, with $\mathbf{u} = (\mathsf{u}_0, \ldots, \mathsf{u}_p)$, as follows:

$$\mathsf{w}_j := M \mathsf{u}_j, \quad j = 0, \ldots, p, \qquad \mathsf{z}_{\pm 1} := \sum_{j=0}^{p} \varphi_j(\pm 1) \mathsf{w}_j, \tag{4.15a}$$

$$(\mathbf{L}\mathbf{u})_j = M A^{-1} \mathsf{w}_j + \frac{\tau^2 \lambda_j}{4} A \mathsf{u}_j + \frac{\tau}{2} \varphi_j(1) \mathsf{z}_1 + \frac{\tau}{2} \varphi_j(-1) \mathsf{z}_{-1}. \tag{4.15b}$$

This approach requires the same number of matrix-vector products as the first approach outlined above, but the communication costs are greatly reduced. Indeed, it is seen that the above procedure requires the parallel computation of the w_j, which are then gathered and reduced by one or two nodes tasked with computing $\mathsf{z}_{\pm 1}$. The results are then broadcast back to all nodes, after which all subsequent computations can be performed in parallel.

Finally, the application of $\mathbf{H}^{-1}$ is trivially parallel, as it requires only that each node solves two linear systems involving the matrix $M+\tau\sqrt{\lambda_j}/2A$, and one application of A. Theorem 3.2 shows that the eigenvalues remain uniformly bounded from above, and that the smaller eigenvalues approach zero as p increases. Therefore, for large p and j, the matrix $M+\tau\sqrt{\lambda_j}/2A$ becomes increasingly similar to M, implying that these systems will become increasingly easy to solve in many applications.

5. Numerical experiments

5.1 *Condition numbers*

In this experiment, we study the dependence of the condition numbers κ of the preconditioned system $\mathbf{H}^{-1}\mathbf{L}$ on the problem and discretization parameters. In order to compute accurately the condition numbers, we consider first a one-dimensional problem. Let M and A be the mass and stiffness matrices obtained by applying piecewise-linear continuous finite elements on a uniform subdivision of the domain $\Omega=(0,1)$ into elements of size $h=2^{-k}$, $k=5,\dots,10$. We note that M and A thus depend on h, although this is left implicit in our notation. For this experiment, we use direct solvers to implement the action of A^{-1} and $(M+\tau\sqrt{\lambda_j}/2A)^{-1}$.

Table 1 shows the condition numbers κ as a function of τ for a wide range of parameters. For this experiment, we set $p=2$ and $h=2^{-5}$. It is found that for either very large or very small τ, $\kappa\to 1$. This is easily explained by the fact that

$$\lim_{\tau\to\infty}\frac{\mathbf{v}^\top\mathbf{L}\mathbf{v}}{\mathbf{v}^\top\mathbf{H}\mathbf{v}}=\lim_{\tau\to 0}\frac{\mathbf{v}^\top\mathbf{L}\mathbf{v}}{\mathbf{v}^\top\mathbf{H}\mathbf{v}}=1. \tag{5.1}$$

Therefore the condition number $\kappa\to 1$ as $\tau\to 0$ or $\tau\to\infty$. Thus the maximal condition number is found for intermediate values of the time-step size τ, although in all cases it satisfies the theoretical bound $\kappa\leqslant 4$ as shown in (3.20). Table 2 shows that the condition number has little to no dependence on mesh refinement, i.e. variation of M and A. As p becomes very large, the condition number approaches an asymptotic value of around $2.686\leqslant 4$, as shown by Table 3; our goal in testing our preconditioners here with very high polynomial degrees is merely to ascertain the asymptotic behaviour of the condition number. The proposed preconditioners are thus fully robust with respect to the parameters, in agreement with the theoretical bound $\kappa\leqslant 4$ from (3.20). Furthermore, this experiment suggests that the efficiency of the preconditioners can exceed theoretical expectations, as shown by condition numbers $\kappa\leqslant 2.686$ throughout these tests.

5.2 *Iteration counts and multigrid preconditioning*

The condition number bound (3.20) holds for the preconditioner $\mathbf{H}$, which assumes that exact solvers are used to apply the inverses of the matrices $M+\tau\sqrt{\lambda_j}/2A$. However, in practice, it is desirable to use

τ	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}	1	10
κ	1.011	1.103	1.749	2.031	2.028	2.019	1.693	1.089

Table 1. Dependence of the condition number κ of $\mathbf{H}^{-1}\mathbf{L}$ on a wide range of values for the time-step size τ. Observe that for either very large or very small τ, the condition number is asymptotically 1, in agreement with (5.1). This explains the observation that the maximum condition number is observed for intermediate values.

κ	$h=2^{-5}$	$h=2^{-6}$	$h=2^{-7}$	$h=2^{-8}$	$h=2^{-9}$	$h=2^{-10}$
$p=1$	1.318	1.319	1.319	1.319	1.319	1.319
$p=2$	2.019	2.019	2.019	2.019	2.019	2.019
$p=3$	2.243	2.243	2.243	2.243	2.243	2.243
$p=4$	2.353	2.353	2.353	2.353	2.353	2.353
$p=5$	2.416	2.417	2.417	2.417	2.417	2.417
$p=6$	2.493	2.493	2.493	2.493	2.493	2.493

Table 2. Dependence of the condition number κ for $p=1,\ldots,6$ and $h=2^{-5},\ldots,2^{-10}$, with $\tau=0.1$. For fixed p, the condition number is insensitive to the mesh size h, i.e. variation of M and A. The condition number slowly increases with p towards its asymptotic value of approximately 2.686, see Table 3.

p	8	16	32	64	128	256
κ	2.558	2.643	2.674	2.684	2.686	2.686

Table 3. Asymptotic behaviour of the condition number with respect to the polynomial degree $p=2^m$, $m=3,\ldots,8$, with fixed $h=2^{-5}$ and $\tau=0.1$. The asymptotic value of κ appears to be $\kappa\approx 2.686$. This suggests that the theoretical bound $\kappa\leqslant 4$ of (3.20) may not be quantitatively sharp in all cases, although it is nevertheless very close.

a cheap approximation, such as a small number of iterations from an iterative solver for $M+\tau\sqrt{\lambda_j}/2A$, which leads to an approximation of the ideal preconditioner $\mathbf{H}^{-1}$.

In this experiment, we study the effect of this approximation, in particular when the inverse of $M+\tau\sqrt{\lambda_j}/2A$ is approximated by a small number of multigrid V-cycles. Let M and A be defined as the mass and stiffness matrices of the piecewise-linear finite element space defined on a uniform triangulation of size $h=2^{-k}$, $k=6,\ldots,10$ in the two-dimensional domain $\Omega=(0,1)^2$. The finest mesh thus leads to more than one million degrees of freedom (DoF) for V. We consider the system $\mathbf{Lu}=\mathbf{g}$, with a chosen exact solution $\mathbf{u}$. We approximate solvers for $M+\tau\sqrt{\lambda_j}/2A$ by applying n V-cycles, $n\in\{1,2,3\}$, with symmetric Gauss–Seidel smoothers. Thus each application of the preconditioner requires $2n$ V-cycles per block. To obtain a fair comparison of all the preconditioners, a relative tolerance of 10^{-6} of the true error in the energy norm was used to determine convergence of PCG, and zero initial guesses were used for all computations.

Table 4 shows the PCG iteration counts for both ideal and approximate preconditioners; in this ex-

Mesh size	DoF	Direct	1 V-cycle	2 V-cycles	3 V-cycles
$h=2^{-6}$	11 907	7	8	7	7
$h=2^{-7}$	48 387	7	8	7	7
$h=2^{-8}$	195 075	7	8	7	7
$h=2^{-9}$	783 363	7	8	7	7
$h=2^{-10}$	3 139 587	7	8	7	7

Table 4. Number of PCG iterations required for convergence of the PCG algorithm in the experiment of section 5.2, for various mesh sizes and corresponding number of degrees of freedom. Fast convergence of PCG is observed in all cases, with the iteration counts being mesh-independent. The method using 1 V-cycle appears as the most efficient in this experiment.

	$h = 2^{-9}$		$h = 2^{-10}$	
Order	Iterations	DoF	Iterations	DoF
$p = 4$	8	1 305 605	8	5 232 645
$p = 6$	9	1 827 847	9	7 325 703
$p = 8$	9	2 350 089	9	9 418 761
$p = 10$	9	2 872 331	9	11 511 819
$p = 12$	9	3 394 573	10	13 604 877
$p = 14$	9	3 916 815	9	15 697 935

Table 5. Dependence of the PCG iteration counts on the polynomial degrees $p = 4, \ldots, 14$ in the experiment of section 5.2, with approximate solvers using 1 multigrid V-cycle. The number of PCG iterations remains uniformly bounded, showing full robustness with respect to the polynomial degree, even up to 15 million DoF. The additional iteration for the case $p = 12$, $h = 2^{-10}$ is explained by the fact that the error on the 9-th iteration was very close to, but above, the convergence criterion.

periment, we set $p = 2$ and $\tau = 0.1$, and thus there are more than 3 million degrees of freedom for $\mathscr{V}_p$ on the finest mesh. Fast and robust convergence of PCG is observed in all cases, showing the effectiveness of these preconditioners. For this experiment, the method using 1 V-cycle appears as the most efficient, as the reduction in cost per iteration outweighs the additional PCG iteration required. In Table 5, we study the dependence of the iteration counts on the polynomial degree. It is found that the use multigrid preconditioners retains the robustness of the preconditioner with respect to the approximation order p. This is indeed the expected result, since Theorem 3.2 shows that as p is increased, for large j the matrices $M + \tau\sqrt{\lambda_j}/2A$ are spectrally closer to M and thus the efficiency of the multigrid V-cycle increases. Our main goal in this experiment is to check the robustness of our preconditioners with respect to the polynomial degree, rather than to actually propose employing high-order approximations in time coupled with low-order approximations in space. Nevertheless, such high-order temporal approximations are encountered in the context of *hp*-version methods (Schötzau & Schwab, 2000; von Petersdorff & Schwab, 2004; Schötzau & Wihler, 2010; Werder *et al.*, 2001). We have also computed the results of this experiment using smaller time-step sizes τ, and we observed a decrease in the iteration counts, as predicted by (5.1). The conclusion drawn from these results is that in practice, standard approximate solvers can be used while retaining the key properties of the ideal preconditioner $\mathbf{H}$, namely the fast convergence of the PCG algorithm and the robustness with respect to the discretization parameters.

5.3 *Parabolic problems with inexact solvers*

The action of $\mathbf{L}$ as given in (4.15) requires the action of A^{-1}. In many practical applications, it is desirable to approximate this step by using an inexact solver, such as a fixed number of multigrid V-cycles, leading to an approximation $\hat{\mathbf{L}} \approx \mathbf{L}$. This raises the question of whether this can be performed without affecting firstly the accuracy of the solution, and secondly the performance of the preconditioning strategy. This experiment provides evidence that inexact solvers can indeed be used without compromising these important objectives.

Consider the piecewise-linear simplicial conforming finite element approximation of the heat equation in the unit square $\Omega = (0,1)^2$, final time $T = 0.1$, imposed with initial datum $u_0(x,y) = x(1-x)\sin(\pi y)$ and with homogeneous Dirichlet lateral boundary conditions. This initial datum is chosen as it leads to a solution with decreased temporal regularity, see (Schötzau & Schwab, 2000). To compare near-exact and inexact solvers, we consider two approaches:

τ/T	Error (BE)	Error (D)	Error (MG)	(D)–(MG)
1	2.546×10^{-2}	3.078×10^{-3}	3.078×10^{-3}	1.303×10^{-8}
1/2	1.475×10^{-2}	3.934×10^{-4}	3.934×10^{-4}	2.129×10^{-8}
1/4	8.008×10^{-3}	5.444×10^{-5}	5.441×10^{-5}	3.219×10^{-8}
1/8	4.178×10^{-3}	8.718×10^{-6}	8.641×10^{-6}	8.126×10^{-8}

Table 6. Final time errors $\|u(T) - u_\tau(T)\|_{L^2(\Omega)}$ obtained by the DG time-stepping method for the problem of section 5.3, for $p = 0$ (the backward Euler method) and for $p = 1$ with direct solvers (D) and inexact multigrid (MG) solvers. The last column gives $\|u_\tau(T) - \hat{u}_\tau(T)\|_{L^2(\Omega)}$, where u_τ, respectively $\hat{u}_\tau$, denotes the solution obtained by (D), respectively (MG).

Iterations	$\tau = T$	$\tau = T/2$	$\tau = T/4$	$\tau = T/8$
(D)	4	3.5	3	3
(MG)	6	5	5	5

Table 7. Average number of PCG iterations per time-step for the problem of section 5.3, with $p = 1$ and with either direct (D) or inexact multigrid (MG) solvers.

(D) direct solvers are used in the application of $\mathbf{P}^\top$, $\mathbf{H}^{-1}$ and $\mathbf{L}$,

(MG) 5 multigrid V-cycles are used to approximate the application of A^{-1} in $\mathbf{L}$ and $\mathbf{P}^\top$, and 1 multigrid V-cycle is used to approximate $(M + \tau\sqrt{\lambda_j}/2A)^{-1}$ in $\mathbf{H}^{-1}$, as in section 5.2.

We point out that the method (MG) does not use any direct solvers throughout the entire computation, and that each PCG iteration costs 7 V-cycles per block.

Table 6 shows the final time errors $\|u(T) - u_\tau(T)\|_{L^2(\Omega)}$, for varying τ, where u denotes the exact solution of the PDE, and where u_τ denotes the discrete solution. Here, we used $h = 2^{-8}$ and $p = 1$. For comparison, we also show the errors attained for $p = 0$, i.e. the backward Euler (BE) method. The inexact solvers for (MG) retain the accuracy of the method, as the difference in solutions between (D) and (MG) is several orders of magnitude smaller than the difference to the exact solution. For $p = 1$, the expected third order super-convergence rate is observed.

Table 7 shows the average number of PCG iterations per time-step required by both approaches in order to obtain a residual tolerance of 10^{-6}, as well as the final time error between the approximate solution u_τ for (D) and $\hat{u}_\tau$ for (MG). This shows that the number of PCG remains robust with respect to the approximation of $\mathbf{L}$ entailed by (MG). This experiment shows that the use of inexact solvers can retain both the accuracy of the DG time-stepping method as well as the efficiency of the proposed preconditioners.

Conclusion

We have developed efficient and robust preconditioners enabling the fast solution of the DG time-stepping method by the preconditioned conjugate gradient algorithm. The analysis and numerical experiments show that the ideal and approximate preconditioners are robust with respect to all discretization parameters and lead to low condition numbers for the preconditioned system. Thus the high-order solution of large problems by the DG time-stepping method is tractable under the proposed approach.

A. Analysis of eigenvalues

In this section, we present the proof of Theorem 3.2. Without loss of generality, it is sufficient to consider only $p \geqslant 2$ throughout this section. We will make use of Weyl's Theorem from eigenvalue perturbation theory (Demmel, 1997).

THEOREM A.1 (Weyl) For a positive integer n, let T and $\hat{T}$ denote symmetric $n \times n$ matrices. Let $\alpha_1 \geqslant \cdots \geqslant \alpha_n$ denote the eigenvalues of T and let $\hat{\alpha}_1 \geqslant \cdots \geqslant \hat{\alpha}_n$ denote the eigenvalues of $\hat{T}$. Then, for each $j = 1, \ldots, n$, we have

$$|\alpha_j - \hat{\alpha}_j| \leqslant \|T - \hat{T}\|_2, \tag{A.1}$$

where $\|\cdot\|_2$ denotes the matrix 2-norm.

We are now ready to prove Theorem 3.2.

Proof of Theorem 3.2. Recall that the matrix T is defined by (4.7). We start by noting that the matrix T is pentadiagonal, i.e. $T_{kj} = 0$ if $|k-j| > 2$. Moreover, it is easy to show from the properties of Legendre polynomials that there exists a constant C independent of p such that

$$\|\psi_k\|_{L^2}^2 = T_{kk} \leqslant C(k+1)^{-2} \quad \forall k = 0, \ldots, p. \tag{A.2}$$

The Cauchy–Schwarz inequality implies that $|T_{jk}| \leqslant \sqrt{T_{kk} T_{jj}}$, and thus all entries of T are uniformly bounded. Since T has at most 5 nonzero entries per row, and all entries are uniformly bounded, the Gershgorin discs of T are bounded independently of p. The Gershgorin Disc Theorem therefore implies that there is a constant C, independent of p, such that λ_0, the maximal eigenvalue of T, satisfies $\lambda_0 \leqslant C$. This corresponds to (3.13) for the case $j = 0$. We consider now $j \geqslant 1$, and without loss of generality, we may assume that $p \geqslant 2$. For $0 \leqslant q \leqslant p-1$, we define the orthogonal projector $\pi_q \colon \mathscr{P}_p \to \mathscr{P}_q$ by

$$\pi_q \colon \sum_{j=0}^{p} \underline{v}_j \psi_j \mapsto \sum_{j=0}^{q} \underline{v}_j \psi_j, \tag{A.3}$$

where the polynomials ψ_j are defined by (4.2), and $\underline{v} = (\underline{v}_0, \ldots, \underline{v}_p) \in \mathbb{R}^{p+1}$. Define the matrix $\hat{T}_q \in \mathbb{R}^{(p+1)\times(p+1)}$ by $(\hat{T}_q)_{kj} := (\pi_q \psi_k, \pi_q \psi_j)_{L^2}$. Note that $(\hat{T}_q)_{kj}$ is zero if either k or j is greater than q, and equals T_{kj} otherwise. Thus, the matrix $\hat{T}_q$ has the general form

$$\hat{T}_q = \begin{bmatrix} \tilde{T} & 0 \\ 0 & 0 \end{bmatrix}, \tag{A.4}$$

where $\tilde{T}$ denotes the $(q+1) \times (q+1)$ principal submatrix of T. The main step of the proof is to use (A.2) repeatedly to find an upper bound for

$$\|T - \hat{T}_q\|_2 = \sup_{\underline{u} \in \mathbb{R}^{p+1} \setminus \{0\}} \sup_{\underline{v} \in \mathbb{R}^{p+1} \setminus \{0\}} \frac{\underline{v}^\top (T - \hat{T}_q) \underline{u}}{\|\underline{u}\|_2 \|\underline{v}\|_2}.$$

For arbitrary $\underline{u}$ and $\underline{v} \in \mathbb{R}^{p+1} \setminus \{0\}$, define the polynomials $u := \sum_{j=0}^{p} \underline{u}_j \psi_j$ and $v = \sum_{j=0}^{p} \underline{v}_j \psi_j$. Then, we have

$$\begin{aligned} \underline{v}^\top (T - \hat{T}_q) \underline{u} &= (u, v)_{L^2} - (\pi_q u, \pi_q v)_{L^2} \\ &= (u - \pi_q u, v - \pi_q v)_{L^2} + (\pi_q u, v - \pi_q v)_{L^2} + (u - \pi_q u, \pi_q v)_{L^2}. \end{aligned} \tag{A.5}$$

It follows from the definition of π_q in (A.3) that

$$\|u-\pi_q u\|_{L^2}^2 = \sum_{k=q+1}^{p}\sum_{j=q+1}^{p} \underline{u}_k\,(\psi_k,\psi_j)_{L^2}\,\underline{u}_j.$$

Since $(\psi_k,\psi_j)_{L^2} = T_{kj} = 0$ if $|k-j|>2$, the strengthened Cauchy–Schwarz inequality yields

$$\sum_{k=q+1}^{p}\sum_{j=q+1}^{p} \underline{u}_k(\psi_k,\psi_j)_{L^2}\underline{u}_j \leqslant 5\left(\sum_{k=q+1}^{p}\|\psi_k\|_{L^2}^2|\underline{u}_j|^2\right) \leqslant \frac{C}{(q+1)^2}\|\underline{u}\|_2^2. \tag{A.6}$$

Note that this implies the following a priori estimate for the orthogonal projector:

$$\|u-\pi_q u\|_{L^2} \leqslant C(q+1)^{-1}\|(\mathscr{I}u)'\|_{L^2} \quad \forall\, u\in\mathscr{P}_p. \tag{A.7}$$

Thus (A.6) and the Cauchy–Schwarz inequality imply that

$$|(u-\pi_q u, v-\pi_q v)_{L^2}| \leqslant C(q+1)^{-2}\|\underline{u}\|_2\|\underline{v}\|_2. \tag{A.8}$$

For $k\leqslant q$ and $j\geqslant q+1$, we have $(\psi_k,\psi_j)=0$ if $k<q-1$ and $j>q+2$, and thus

$$(\pi_q u, v-\pi_q v)_{L^2} = \sum_{k=0}^{q}\sum_{j=q+1}^{p}\underline{u}_k(\psi_k,\psi_j)\underline{v}_j = \sum_{k=\max(q-1,0)}^{q}\sum_{j=q+1}^{\min(p,q+2)}\underline{u}_k(\psi_k,\psi_j)\underline{v}_j.$$

The Cauchy–Schwarz inequality thus implies that

$$\begin{aligned}|(\pi_q u, v-\pi_q v)_{L^2}| &\leqslant 2\left(\sum_{k=\max(q-1,0)}^{q}|\underline{u}_k|^2\|\psi_k\|_{L^2}^2\right)^{\frac12}\left(\sum_{j=q+1}^{\min(p,q+2)}|\underline{v}_j|^2\|\psi_j\|_{L^2}^2\right)^{\frac12}\\ &\leqslant C(q+1)^{-2}\|\underline{u}\|_2\|\underline{v}\|_2.\end{aligned} \tag{A.9}$$

An identical argument shows that

$$|(u-\pi_q u, \pi_q v)_{L^2}| \leqslant C(q+1)^{-2}\|\underline{u}\|_2\|\underline{v}\|_2. \tag{A.10}$$

Combining (A.8), (A.9) and (A.10) implies that there exists a constant C, independent of p and q, such that

$$\|T-\hat{T}_q\|_2 \leqslant C(q+1)^{-2}. \tag{A.11}$$

Therefore, Weyl's Theorem implies that the eigenvalues λ_j of T and $\hat{\lambda}_j$ of $\hat{T}_q$ satisfy

$$|\lambda_j-\hat{\lambda}_j| \leqslant C(q+1)^{-2} \quad \forall\, j=0,\ldots,p, \tag{A.12}$$

where the constant C is independent of j, q and p. However, it is clear from (A.4) that $\hat{\lambda}_j=0$ for $j\geqslant q+1$, and thus there exists a constant C independent of p and q such that

$$\lambda_j = |\lambda_j-\hat{\lambda}_j| \leqslant C(q+1)^{-2} \quad \forall\, j\geqslant q+1. \tag{A.13}$$

The left-hand side of this inequality is independent of q while the right-hand side is valid of all $q\leqslant j-1$. Therefore, the choice $q=j-1$ yields (3.13) for $j=1,\ldots,p$.

The lower bound (3.14) follows from inverse inequalities. Indeed, λ_p satisfies

$$\lambda_p = \min_{v \in \mathscr{P}_p \setminus \{0\}} \frac{\|v\|_{L^2}^2}{\|(\mathscr{I}v)'\|_{L^2}^2}.$$

To obtain (3.14), it is therefore enough to show the inverse inequality

$$\|(\mathscr{I}v)'\|_{L^2} \leqslant C(p+1)^2 \|v\|_{L^2} \quad \forall v \in \mathscr{P}_p. \tag{A.14}$$

For any $v \in \mathscr{P}_p$, standard inverse inequalities (Schwab, 1998) imply that

$$\begin{aligned} \|(\mathscr{I}v)'\|_{L^2} &\leqslant \|v'\|_{L^2} + \|v\|_{L_\infty \frac{1}{2}} \|(L_p - L_{p+1})'\|_{L^2} \\ &\lesssim \left((p+1)^2 + (p+1)\,\|(L_p - L_{p+1})'\|_{L^2}\right) \|v\|_{L^2}. \end{aligned} \tag{A.15}$$

It follows from (4.10) that

$$\|(L_p - L_{p+1})'\|_{L^2}^2 = \sum_{j=0}^{p} \frac{8(j+\frac{1}{2})^2}{2j+1} = 2\,(p+1)^2. \tag{A.16}$$

Substituting (A.16) into (A.15) yields (A.14). □

Acknowledgements

The author wishes to express his thanks to Paul Houston, Lorenz John, Christian Kreuzer, Endre Süli and Martin Vohralík for many helpful discussions. The author was supported by an EPSRC Doctoral Prize at the Mathematical Institute, University of Oxford, during the period in which part of this work was completed.

REFERENCES

AKRIVIS, G., MAKRIDAKIS, C. & NOCHETTO, R. H. (2009) Optimal order a posteriori error estimates for a class of Runge-Kutta and Galerkin methods. *Numer. Math.*, **114**, 133–160.

AKRIVIS, G. & MAKRIDAKIS, C. (2004) Galerkin time-stepping methods for nonlinear parabolic equations. *M2AN Math. Model. Numer. Anal.*, **38**, 261–289.

AXELSSON, O. (1969) A class of *A*-stable methods. *Nordisk Tidskr. Informationsbehandling (BIT)*, **9**, 185–199.

BICKART, T. A. (1977) An efficient solution process for implicit Runge-Kutta methods. *SIAM J. Numer. Anal.*, **14**, 1022–1027.

BUTCHER, J. C. (1976) On the implementation of implicit Runge-Kutta methods. *Nordisk Tidskr. Informationsbehandling (BIT)*, **16**, 237–240.

CHRYSAFINOS, K. & WALKINGTON, N. J. (2006) Error estimates for the discontinuous Galerkin methods for parabolic equations. *SIAM J. Numer. Anal.*, **44**, 349–366 (electronic).

DELFOUR, M., HAGER, W. & TROCHU, F. (1981) Discontinuous Galerkin methods for ordinary differential equations. *Math. Comp.*, **36**, 455–473.

DEMMEL, J. W. (1997) *Applied numerical linear algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, pp. xii+419.

ERIKSSON, K., JOHNSON, C. & THOMÉE, V. (1985) Time discretization of parabolic problems by the discontinuous Galerkin method. *RAIRO Modél. Math. Anal. Numér.*, **19**, 611–643.

ERIKSSON, K. & JOHNSON, C. (1991) Adaptive finite element methods for parabolic problems. I. A linear model problem. *SIAM J. Numer. Anal.*, **28**, 43–77.

ERIKSSON, K. & JOHNSON, C. (1995) Adaptive finite element methods for parabolic problems. II. Optimal error estimates in $L_\infty L_2$ and $L_\infty L_\infty$. *SIAM J. Numer. Anal.*, **32**, 706–740.

GAUTSCHI, W. (1997) *Numerical analysis*. Birkhäuser Boston, Inc., Boston, MA, pp. xiv+506. An introduction.

GRADSHTEYN, I. S. & RYZHIK, I. M. (2015) *Table of integrals, series, and products*, eighth edition, Elsevier/Academic Press, Amsterdam, pp. xlvi+1133. Translated from the Russian, Translation edited and with a preface by Daniel Zwillinger and Victor Moll, Revised from the seventh edition.

HAIRER, E. & WANNER, G. (2010) *Solving ordinary differential equations. II*. Springer Series in Computational Mathematics, vol. 14. Berlin: Springer-Verlag, pp. xvi+614. Stiff and differential-algebraic problems, Second revised edition, paperback.

HESTENES, M. R. & STIEFEL, E. (1952) Methods of conjugate gradients for solving linear systems. *J. Research Nat. Bur. Standards*, **49**, 409–436 (1953).

HIPTMAIR, R. (2006) Operator preconditioning. *Comput. Math. Appl.*, **52**, 699–706.

HULME, B. L. (1972) Discrete Galerkin and related one-step methods for ordinary differential equations. *Math. Comp.*, **26**, 881–891.

JAMET, P. (1978) Galerkin-type approximations which are discontinuous in time for parabolic equations in a variable domain. *SIAM J. Numer. Anal.*, **15**, 912–928.

MAKRIDAKIS, C. G. & BABUŠKA, I. (1997) On the stability of the discontinuous Galerkin method for the heat equation. *SIAM J. Numer. Anal.*, **34**, 389–401.

MAKRIDAKIS, C. & NOCHETTO, R. H. (2006) A posteriori error analysis for higher order dissipative methods for evolution problems. *Numer. Math.*, **104**, 489–514.

MÁLEK, J. & STRAKOŠ, Z. (2015) *Preconditioning and the conjugate gradient method in the context of solving PDEs*. SIAM Spotlights, vol. 1. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, pp. x+104.

MARDAL, K.-A., NILSSEN, T. K. & STAFF, G. A. (2007) Order-optimal preconditioners for implicit Runge-Kutta schemes applied to parabolic PDEs. *SIAM J. Sci. Comput.*, **29**, 361–375 (electronic).

PEARSON, J. W. & WATHEN, A. J. (2012) A new approximation of the Schur complement in preconditioners for PDE-constrained optimization. *Numer. Linear Algebra Appl.*, **19**, 816–829.

RICHTER, T., SPRINGER, A. & VEXLER, B. (2013) Efficient numerical realization of discontinuous Galerkin methods for temporal discretization of parabolic problems. *Numer. Math.*, **124**, 151–182.

SCHÖTZAU, D. & SCHWAB, C. (2000) Time discretization of parabolic problems by the hp-version of the discontinuous Galerkin finite element method. *SIAM J. Numer. Anal.*, **38**, 837–875.

SCHÖTZAU, D. & WIHLER, T. P. (2010) A posteriori error estimation for hp-version time-stepping methods for parabolic partial differential equations. *Numer. Math.*, **115**, 475–509.

SCHWAB, C. (1998) *p- and hp-finite element methods*. Numerical Mathematics and Scientific Computation. New York: The Clarendon Press Oxford University Press, pp. xii+374.

THOMÉE, V. (2006) *Galerkin finite element methods for parabolic problems*. Springer Series in Computational Mathematics, vol. 25, second edn. Berlin: Springer-Verlag, pp. xii+370.

VON PETERSDORFF, T. & SCHWAB, C. (2004) Numerical solution of parabolic equations in high dimensions. *M2AN Math. Model. Numer. Anal.*, **38**, 93–127.

WATHEN, A. J. (2015) Preconditioning. *Acta Numer.*, **24**, 329–376.

WELLER, S. & BASTING, S. (2015) Efficient preconditioning of variational time discretization methods for parabolic partial differential equations. *ESAIM Math. Model. Numer. Anal.*, **49**, 331–347.

WERDER, T., GERDES, K., SCHÖTZAU, D. & SCHWAB, C. (2001) hp-discontinuous Galerkin time stepping for parabolic problems. *Comput. Methods Appl. Mech. Engrg.*, **190**, 6685–6708.

WLOKA, J. (1987) *Partial differential equations*. Cambridge: Cambridge University Press, pp. xii+518. Translated from the German by C. B. Thomas and M. J. Thomas.