



Wavelet-in-time multigrid-in-space preconditioning of parabolic evolution equations

Roman Andreev

► To cite this version:

Roman Andreev. Wavelet-in-time multigrid-in-space preconditioning of parabolic evolution equations. SIAM Journal on Scientific Computing, 2016, 10.1137/140998639 . hal-01184494

HAL Id: hal-01184494

<https://hal.science/hal-01184494>

Submitted on 14 Aug 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

WAVELET-IN-TIME MULTIGRID-IN-SPACE PRECONDITIONING OF PARABOLIC EVOLUTION EQUATIONS*

ROMAN ANDREEV†

Abstract. Two space-time variational formulations of linear parabolic evolution equations are considered, one is symmetric and elliptic on the trial space while the other is not. In each case, a space-time Petrov–Galerkin discretization using suitable tensor product trial and test functions leads to a large linear system of equations. The well-posedness of this system with respect to parabolic norms induces a canonical preconditioner for the algebraic equations that arise after a choice of basis. For the iterative resolution of this algebraic system with parallelization in the temporal direction we propose a sparse algebraic wavelet-in-time transformation on possibly nonuniform temporal meshes. This transformation approximately block-diagonalizes the preconditioner, and the individual spatial blocks can then be inverted for instance by standard spatial multigrid methods in parallel. The performance of the preconditioner is documented in a series of numerical experiments.

Key words. space-time, parabolic, discretization, preconditioning, wavelet, multigrid

AMS subject classifications. 35K90, 65M12, 65M20, 65M55, 65M60, 65Y05

1. Introduction and problem formulation.

1.1. Introduction. We present a preconditioner for space-time simultaneous Petrov–Galerkin discretizations of linear parabolic evolution equations that is suitable for large scale parallel-in-time computation. The basic rationale for the design of the preconditioner is the ability to use existing spatial finite element codes and preconditioners for elliptic problems, such as multigrid. In the context of large scale parallel computation we therefore stipulate that the space-time solution vector is distributed across computing nodes along the temporal dimension so that a node, or a group of nodes, each hold only a few temporal snapshots of the solution out of many. By many we mean hundreds or thousands of snapshots. Then, space-time simultaneous iterative computation asks for the reduction of communication between these groups of nodes as far as possible, in particular when it comes to preconditioning. This will be achieved by a transformation to a temporal multiscale basis.

We consider two different space-time variational formulations of the linear parabolic evolution equation (Section 1.3). The key difference between them is that one is symmetric and elliptic while the other is not. The lack of symmetry has implications on the choice of discrete trial and test spaces and on the choice of the iterative solver (Sections 2.1–2.4). The symmetric formulation, on the other hand, involves the inverse of the spatial operator, which renders its numerical implementation more challenging. However, we will show that both formulations admit the same canonical preconditioner induced by the continuous norm on the trial space (Section 2.4). Throughout, we assume the regime of small Péclet numbers. That is we assume that the symmetric part of the generator is elliptic, and the asymmetric part is relatively small.

As we shall explain (Section 3.2.2), the space-time preconditioner involves the construction of a temporal piecewise polynomial multiscale basis with compact support which can be rescaled to a Riesz basis both in the Lebesgue space L_2 and in the Sobolev space H^1 . Many such bases of wavelet type are known, but their construction

*First version: RICAM report 2014-27, December 2, 2014. This version: August 14, 2015

†Univ Paris Diderot, Sorbonne Paris Cité, LJLL (UMR 7598 CNRS), F-75205 Paris, France (roman.andreev@upmc.fr)

is typically based on dyadic refinement, posing restrictions on the temporal mesh. The classical fast Fourier transform suffers from the same drawback. For piecewise linear spline spaces we provide an algebraic algorithm that works on arbitrary temporal meshes (Section 3.3). The algorithm proceeds as follows: a) identify basis functions in the top frequency band; b) approximately orthogonalize to those of lower frequency; c) split off the resulting detail space; d) repeat. Approximate iterative orthogonalization in the second step allows to produce compactly supported wavelet-like functions. Due to the purely algebraic approach no complications appear at the boundary of the interval. Extensions to splines of higher polynomial degree are of interest but will not be discussed here.

The sparsity of the multiscale transformation to the wavelet-like basis in the temporal variable is *crucial* for reducing the interprocess communication cost. Moreover, the Riesz basis property simultaneously in L_2 and H^1 allow to transform and approximate the nonlocal canonical preconditioner as a block-diagonal one, where each block corresponds to a spatial Helmholtz problem with an imaginary frequency. For the purpose of preconditioning, this block can be replaced by multigrid versions of positive definite Helmholtz problems with real frequency while maintaining control over the quality of the preconditioner (Section 3.2). This block-diagonal preconditioner can then be applied in parallel. We emphasize that parallelization is achieved in the temporal direction, but the effect of the multiscale wavelet-like transformation in time is such that the individual spatial blocks of the preconditioner do not correspond to small temporal subintervals, rather to wave packets of different temporal scale.

In Section 4 we present our numerical experiments that document the quality of the preconditioner and its performance in the space-time simultaneous resolution of parabolic evolution equations. We conclude in Section 5.

In the Appendix we provide A) a Matlab code for the construction of the wavelet-like basis, and B) a generalized LSQR algorithm that is used in the complete space-time algorithm.

1.2. Problem class. Let V and H be real separable Hilbert spaces with continuous and dense embedding $V \hookrightarrow H$. The Hilbert space H is identified with its (continuous) dual H' by the Riesz isomorphism. This results in the Gelfand triple $V \hookrightarrow H \cong H' \hookrightarrow V'$ with continuous and dense embeddings. We write $\|\cdot\|_V$ for the norm on V . The scalar product on H and (its continuous extension to) the duality pairing between V and V' is denoted by $(\cdot, \cdot)$. For the duality pairing on other Banach space we write $\langle \cdot, \cdot \rangle$. We write $L_2(J; V')$ for the Bochner space of V' -measurable functions on J , and $H^1(J; V')$ for the Bochner–Sobolev subspace of weakly differentiable functions w on J with derivative $d_t w$ in $L_2(J; V')$, see [22] or [39, Chapter 1]. Similar notation is employed for other instances of Hilbert spaces.

Let $J = (0, T)$ be a nonempty bounded interval. Let $A(t) : V \rightarrow V'$, (a.e.) $t \in J$, be a family of bounded linear operators. We assume that, for some fixed constants $C > 0$, $\alpha > 0$, and $\gamma_0 \geq 0$, the family $A(t)$, $t \in J$, satisfies the conditions

$$\text{a) For all } \chi, \tilde{\chi} \in V, \text{ the mapping } t \mapsto (A(t)\chi, \tilde{\chi}) \text{ is Lebesgue measurable.} \quad (1)$$

$$\text{b) } |(A(t)\chi, \tilde{\chi})| \leq C \|\chi\|_V \|\tilde{\chi}\|_V \text{ for all } \chi, \tilde{\chi} \in V \text{ and (a.e.) } t \in J. \quad (2)$$

$$\text{c) } (A(t)\chi, \chi) \geq \alpha^2 \|\chi\|_V^2 - \gamma_0^2 \|\chi\|_H^2 \text{ for all } \chi \in V \text{ and (a.e.) } t \in J. \quad (3)$$

Let $f \in L_2(J; V')$ and $g \in H$ be given. The subject of this paper is the abstract linear parabolic evolution equation

$$u(0) = g, \quad d_t u(t) + A(t)u(t) = f(t) \quad (\text{a.e.}) \quad t \in J, \quad (4)$$

where the equalities are understood in H and V' , respectively. Equivalently, one could write $d_t u + Au = f$ with equality in $L_2(J; V')$, where Au stands for the map $t \mapsto A(t)u(t)$, $t \in J$. In order to make the concept of a solution precise, we introduce the Sobolev–Bochner spaces

$$X := L_2(J; V) \cap H^1(J; V') \quad \text{and} \quad Y := L_2(J; V) \times H. \quad (5)$$

A generic element of Y has the form $v = (v_1, v_0)$. A norm on X is given by

$$\|w\|_X^2 := \|w\|_{L_2(J; V')}^2 + \|d_t w\|_{H^1(J; V')}^2 \quad w \in X. \quad (6)$$

We recall the continuous embedding $X \hookrightarrow C^0([0, T]; H)$. In particular, $w(t_0)$ is well-defined in H for any $w \in X$ and $0 \leq t_0 \leq T$, and $w \mapsto w(t_0)$ is continuous, see [22, Section 5.9.2] or [39, Chapter 1]. An equivalent norm (with equivalence constants depending on T , see Section 2.5) is therefore given by

$$\|w\|_X^2 := \|w\|_X^2 + \|w(T)\|_H^2, \quad w \in X, \quad (7)$$

and henceforth, X is understood to be equipped with this norm. With the obvious corresponding inner product, it is a Hilbert space. The choice of the norm can be motivated as follows. Suppose A is the embedding $V \hookrightarrow V'$. Then $\|d_t u + Au\|_{L_2(J; V')}^2 = \|u\|_X^2 - \|u(0)\|_H^2$ can be verified by expanding the square and using integration by parts. Therefore $\|u\|_X^2 = \|f\|_{L_2(J; V')}^2 + \|g\|_H^2$ if u satisfies (4).

Under the stated assumptions on A and the data f and g , there exists a unique solution $u \in X$ to (4), and it depends continuously on the data [39, Chapter 3, §4.7].

In view of the usual transformation $u \mapsto v := [t \mapsto u(t)e^{-\gamma_0^2 t}]$, which is an automorphism on X , we assume from now on without loss of generality that $\gamma_0 = 0$.

1.3. Space-time variational formulations. Our first space-time variational formulation is from [46]. It is based on testing $d_t u + Au = f$ by $v_1 \in L_2(J; V')$ and appending the initial condition $u(0) = g$ using a Lagrange multiplier $v_0 \in H$. Specifically, define the bounded linear operator $B : X \rightarrow Y'$ by

$$\langle Bw, v \rangle := \int_J (d_t w + Aw, v_1) dt + (w(0), v_0), \quad (w, v) \in X \times Y, \quad (8)$$

and the bounded linear functional $F \in Y'$ by

$$Fv := \int_J (f, v_1) dt + (g, v_0), \quad v \in Y. \quad (9)$$

Here and in the following we omit the dependence of the integrands on t . The space-time variational formulation then reads

$$\text{Find } u \in X : \quad \langle Bu, v \rangle = Fv \quad \forall v \in Y. \quad (10)$$

Our second variational formulation is an instance of the Brézis–Ekeland–Nayroles variational principle [15, 41]. To state it, we introduce the notations

$$\hat{A} := \frac{1}{2}(A + A'), \quad \tilde{A} := \frac{1}{2}(A - A') \quad \text{and} \quad \tilde{C} := d_t w + \tilde{A}w, \quad w \in X, \quad (11)$$

where $\hat{A}$ and $\tilde{A}$ denote the symmetric and the anti-symmetric part of A , respectively. Define the bounded linear operator $\hat{B} : X \rightarrow X'$ by

$$\langle \hat{B}w, v \rangle := \int_J \left\{ (\hat{A}^{-1} \tilde{C}w, \tilde{C}v) + (\hat{A}w, v) \right\} dt + (w(T), v(T)), \quad (12)$$

for $w, v \in X$, and the bounded linear functional $\widehat{F} \in X'$ by

$$\widehat{F}v := \int_J (f, \widehat{A}^{-1}\widetilde{C}v + v)dt + (g, v(0)), \quad v \in X. \quad (13)$$

The second space-time variational formulation then reads

$$\text{Find } \widehat{u} \in X : \quad \langle \widehat{B}\widehat{u}, w \rangle = \widehat{F}w \quad \forall w \in X. \quad (14)$$

The essential property of this space-time variational formulation to (10) is that the trial space and the test space coincide, and the operator $\widehat{B}$ is symmetric, that is $\langle \widehat{B}w, v \rangle = \langle \widehat{B}v, w \rangle$ for all $w, v \in X$. Moreover, owing to X -ellipticity of $\widehat{B}$,

$$\exists \widehat{\gamma} > 0 : \quad \langle \widehat{B}w, w \rangle \geq \widehat{\gamma}^2 \|w\|_X^2 \quad \forall w \in X, \quad (15)$$

see [2, Proposition 3.2.26], the Lax–Milgram lemma shows that there is a unique solution $\widehat{u}$ to (14), and it depends continuously on the data. It is instructive and straightforward to verify that u from (4) satisfies the variational formulation (14).

For completeness, we mention another space-time variational formulation [7, 8, 27, 16, 40] which is based on testing $d_t u + Au = f$ by $v \in X$ with $v(T) = 0$, performing integration by parts in time, and putting the adjoint of A onto the test function. This exposes $u(0)$, which is then replaced by the known initial datum g . The result is the space-time variational formulation: Find $u \in L_2(J; V)$ such that $\int_J (u, -d_t v + A'v)dt = \int_J (f, v)dt + (g, v(0))$ for all $v \in X$ with $v(T) = 0$. It can be treated analogously to the first one. Yet another space-time variational formulation based on fractional temporal derivatives was given in [37], with the effect that only H and V appear in the definitions of the trial and test spaces, but not V' . Finally, time-periodic problems with periodicity condition $u(0) = u(T)$ require only minor modifications, and will not be discussed here.

In the remainder of the paper we focus entirely on space-time full tensor product discretizations of the variational formulations (10) and (14). We point out, however, that they can serve as a basis for space-time adaptive and space-time compressive discretizations. Some references in that direction are given in the next section.

1.4. A brief overview of space-time methods. In order to improve on time-stepping schemes and to achieve parallelization in time, several methods have been developed for parabolic evolution equations. Without claiming exhaustiveness we mention waveform relaxation [50, 33, 31], low rank tensor approximation [21], parallel multiple shooting [35] and “parareal” [38, 25], Laplace transformation [47], Toeplitz matrix representation [9], space-time multigrid [29, 32, 24], unstructured space-time decomposition [42], space-time wavelet discretization [1] and sparse grids [28, 27] with heuristic adaptivity. See [23] for a historical perspective and further references.

Those works do not exploit the fact that parabolic evolution equations define well-posed operator equations between Bochner spaces, and thus admit well-posed space-time variational formulations. This was first done in [7, 8] using conforming discretizations of a space-time variational formulation of the heat equation with piecewise polynomials in time. In [46] and subsequently [16, 17, 34, 37], the applicability of adaptive wavelet methods to parabolic evolution equations was shown. By means of a reduction to a boundary integral equation [20], stable space-time compressive algorithms were constructed in [18]. Well-posed space-time variational formulations and stable a priori Petrov–Galerkin discretizations were derived and implemented in [36, 40, 4, 5], space-time compressive variants in [2, 3, 6]. Quasi-optimality in space-time norms and space-time adaptivity were investigated in [49].

2. Conforming space-time tensor product discretization.

2.1. Space-time tensor product subspaces. The discrete space-time trial and test spaces are built from nested finite-dimensional univariate temporal subspaces $E_L \subset H^1(J)$, $F_L \subset L_2(J)$, and spatial subspaces $V_L \subset V$ parameterized by an integer $L \geq 0$. We then define the discrete trial and test spaces (recall X and Y from (5))

$$X_L := E_L \otimes V_L \subset X \quad \text{and} \quad Y_L := (F_L \otimes V_L) \times V_L \subset Y, \quad (16)$$

where $\otimes$ denotes the algebraic tensor product. If $\bigcup_{L \geq 0} E_L$ is dense in $H^1(J)$ and $\bigcup_{L \geq 0} V_L$ is dense in V then $\bigcup_{L \geq 0} X_L$ is dense in X .

We assume henceforth that all subspaces are nontrivial and satisfy the discrete inf-sup condition

$$\forall L \geq 0 : \quad \gamma_L := \inf_{w \in X_L \setminus \{0\}} \sup_{v \in Y_L \setminus \{0\}} \frac{\langle Bw, v \rangle}{\|w\|_X \|v\|_Y} > 0, \quad (17)$$

where B is as in (8). Note that at this stage, $\gamma_L > 0$ may not be bounded away from zero as $L \rightarrow \infty$.

By ellipticity (15) of the symmetric operator $\hat{B}$, its discrete inf-sup constant $\hat{\gamma}_L$ automatically satisfies $\hat{\gamma}_L \geq \hat{\gamma} > 0$ when X_L is used simultaneously as the trial and as the test space.

In the quantification of the well-posedness of the discrete problems the quantity

$$\Gamma_L := \sup_{w \in X_L \setminus \{0\}} \sup_{v \in Y_L \setminus \{0\}} \frac{\langle Bw, v \rangle}{\|w\|_X \|v\|_Y}, \quad L \geq 0, \quad (18)$$

will also play a role. It is clear that $\Gamma_L \leq \|B\|$. Similarly, the corresponding supsup quantity $\hat{\Gamma}_L$ for $\hat{B}$ satisfies $\hat{\Gamma}_L \leq \|\hat{B}\|$.

The discrete inf-sup condition (17) necessitates $\dim X_L \leq \dim Y_L$. We discuss the cases $\dim X_L = \dim Y_L$ and $\dim X_L < \dim Y_L$ in Sections 2.2 and 2.3, respectively.

2.2. Discrete variational formulation. In this section we assume that $X_L \subset X$ and $Y_L \subset Y$ have the same dimension. The existence of such subspaces satisfying the discrete inf-sup condition (17) is guaranteed by bounded invertibility of $B : X \rightarrow Y'$. Practical instances are more subtle due to the non-symmetric contribution of the temporal derivative; however, in [5] it was shown that collocation Runge–Kutta time-stepping schemes applied to spatial semi-discretizations admit an interpretation as a discrete space-time variational formulation, see (19) below, with discrete trial and test spaces of the form (16). In addition, discrete stability $\gamma_L \geq \gamma > 0$ in (17) may be achieved for all $L \geq 0$ under mild assumptions on the discretization parameters.

Given such X_L and Y_L , it is straightforward to define the discrete solution by the discrete nonsymmetric space-time variational formulation

$$\text{Find } u_L \in X_L : \quad \langle Bu_L, v \rangle = Fv \quad \forall v \in Y_L. \quad (19)$$

With the assumption that the discrete inf-sup condition (17) holds for X_L and Y_L , a unique solution $u_L \in X_L$ to (19) exists by standard finite element method theory.

For any nontrivial subspace $X_L \subset X$, symmetry and ellipticity (15) of $\hat{B}$ guarantee its ellipticity on X_L . The Lax–Milgram theorem then provides a unique solution to the discrete symmetric space-time variational formulation

$$\text{Find } \hat{u}_L \in X_L : \quad \langle \hat{B}\hat{u}_L, w \rangle = \hat{F}w \quad \forall w \in X_L. \quad (20)$$

Again, we emphasize that X_L is used as the trial and as the test space simultaneously. We will content ourselves with the full tensor product subspace X_L as in (16), but we mention that space-time compressive discretizations X_L and Y_L could be used for (19) and for (20), see [3].

Let Φ be a basis for X_L , and Ψ a basis for Y_L . Here, and where appropriate, we omit the subscript L for readability. We write $u_L = \Phi^\top \mathbf{u}$ with a vector of coefficients $\mathbf{u} \in \mathbb{R}^\Phi$ indexed by the elements of Φ , and define the system matrix $\mathbf{B} \in \mathbb{R}^{\Psi \times \Phi}$ with the components $\mathbf{B}_{\psi\phi} = \langle B\phi, \psi \rangle$, as well as the load vector $\mathbf{F} \in \mathbb{R}^\Psi$ with the components $\mathbf{F}_\psi = F\psi$, where $(\phi, \psi) \in \Phi \times \Psi$. Then the discrete variational formulation (19) is equivalent to the algebraic equation

$$\mathbf{B}\mathbf{u} = \mathbf{F}. \quad (21)$$

The discrete inf-sup condition (17) implies that $\mathbf{B}$ is injective, so that (21), and also (24) below, is uniquely solvable.

Defining $\hat{\mathbf{B}} \in \mathbb{R}^{\Phi \times \Phi}$ and $\hat{\mathbf{F}} \in \mathbb{R}^\Phi$ analogously, and writing $\hat{u}_L := \Phi^\top \hat{\mathbf{u}}$, the symmetric discrete variational formulation (20) is equivalent to the algebraic equation

$$\hat{\mathbf{B}}\hat{\mathbf{u}} = \hat{\mathbf{F}}. \quad (22)$$

2.3. Minimal residual variational formulation. For details and proofs for this section we refer to [3, 2]. Contrary to the foregoing section we now assume that $\dim Y_L \geq \dim X_L$, both finite but possibly unequal. The motivation to consider this case is the fact that the discrete inf-sup condition (17) and indeed, discrete stability $\gamma_L \geq \gamma > 0$, is then easier to achieve, for one is allowed to choose any discrete test space Y_L that is “large enough”. In this case the discrete variational formulation (19) is meaningless. Instead, we define the discrete solution as the minimizer of the functional residual by

$$u_L := \operatorname{argmin}_{w_L \in X_L} R(w_L), \quad R(w_L) := \sup_{v \in Y_L \setminus \{0\}} \frac{|\langle F - Bw_L, v \rangle|}{\|v\|_Y}. \quad (23)$$

In the particular case $\dim X_L = \dim Y_L$ this formulation reduces to (19), so it is appropriate to use the same symbol u_L here. Under the inf-sup condition (17) there exists a unique solution $u_L \in X_L$ to (23), it depends linearly and continuously on the load functional F , and the operator norm of the discrete solution mapping $F \mapsto u_L$ is bounded by $1/\gamma_L$. Defining $\mathbf{B}$, $\mathbf{F}$, and $\mathbf{u}$ as in the foregoing section, the functional residual minimization (23) is equivalent to the algebraic equation

$$\mathbf{B}^\top \mathbf{N}^{-1} \mathbf{B} \mathbf{u} = \mathbf{B}^\top \mathbf{N}^{-1} \mathbf{F}, \quad (24)$$

where $\mathbf{N}$ is the symmetric positive definite matrix such that $\mathbf{v}^\top \mathbf{N} \mathbf{v} = \|\Psi^\top \mathbf{v}\|_Y^2$ for all coefficient vectors $\mathbf{v} \in \mathbb{R}^\Psi$. Note that (24) has the form of generalized Gauss normal equations. If $\dim X_L = \dim Y_L$ then it is equivalent to (21), and therefore henceforth we consider (21) as a special case of (24), and do not discuss it separately. We will solve (24) iteratively with the generalized LSQR algorithm provided in Appendix B.

Instead of the exact $\mathbf{N}$ in (24), an approximation may be used, say $\mathbf{N}'$ with $c_{\mathbf{N}}^2 \mathbf{N} \leq \mathbf{N}' \leq C_{\mathbf{N}}^2 \mathbf{N}$ for some constants $0 < c_{\mathbf{N}} \leq C_{\mathbf{N}} < \infty$. If $\mathbf{u}'$ denotes the corresponding solution, and $u'_L := \Phi^\top \mathbf{u}'$ then the quasi-optimality estimate holds:

$$\|u - u'_L\|_X \leq C_L \inf_{w_L \in X_L} \|u - w_L\|_X \quad \text{with} \quad C_L := \frac{C_{\mathbf{N}} \Gamma_L}{c_{\mathbf{N}} \gamma_L}. \quad (25)$$

Our objective now is to devise a preconditioner for the symmetric algebraic equations (22) and (24). This is the subject of the next section.

2.4. Parabolic space-time preconditioners. As in the foregoing Sections 2.2 and 2.3, assume $X_L \subset X$ and $Y_L \subset Y$ are nontrivial finite-dimensional subspaces that satisfy the discrete inf-sup condition (17), and with $\dim X_L \leq \dim Y_L$ possibly unequal. The algebraic equations (22) and (24) stem from a space-time Petrov–Galerkin discretization of a parabolic evolution equation, so that direct solution is likely impossible due to the large size of the system. Two basic methods for their approximate iterative solution are therefore

- the preconditioned Richardson iteration, and
- the preconditioned conjugate gradients method.

It may be preferable to apply the Richardson iteration to the symmetric formulation (22) due to the fact that even the forward application on $\hat{\mathbf{B}}$ requires solutions of elliptic problems. The preconditioned conjugate gradient method applied to the normal equations (24) is equivalent to the generalized LSQR method given in the Appendix.

Both iterative schemes necessitate a good preconditioner, that is a matrix $\mathbf{M}$ such that the condition numbers (27) are small (or the eigenvalues are clustered) [26]. To exhibit such a preconditioner we exploit the mapping properties of the operators B and $\hat{B}$. Given a basis $\Phi \subset X_L$ we define $\mathbf{M} \in \mathbb{R}^{\Phi \times \Phi}$ as the matrix such that $\mathbf{w}^\top \mathbf{M} \mathbf{w} = \|\Phi^\top \mathbf{w}\|_X^2$ for all coefficient vectors $\mathbf{w}$. Its components are then

$$\mathbf{M}_{\varphi\phi} = \int_J \{(\phi(t), \varphi(t))_V + (d_t \phi(t), d_t \varphi(t))_{V'}\} dt + (\phi(T), \varphi(T)). \quad (26)$$

The variational characterization of the extremal singular values shows that the set of singular values σ of the preconditioned matrices satisfies $\sigma(\mathbf{M}^{-1/2} \hat{\mathbf{B}} \mathbf{M}^{-1/2}) \subset [\hat{\gamma}_L, \hat{\Gamma}_L]$ and $\sigma(\mathbf{N}^{-1/2} \mathbf{B} \mathbf{M}^{-1/2}) \subset [\gamma_L, \Gamma_L]$, and the intervals are the smallest possible, see [2, Proof of Proposition 4.2.3], and Section 2.2 for the definition of these quantities. In practice, only $\mathbf{M}^{-1}$ and $\mathbf{N}^{-1}$ will be needed, but not their square roots. It follows

$$\kappa_2(\mathbf{M}^{-1/2} \hat{\mathbf{B}} \mathbf{M}^{-1/2}) = \hat{\Gamma}_L / \hat{\gamma}_L \quad \text{and} \quad \kappa_2(\mathbf{M}^{-1/2} \mathbf{B}^\top \mathbf{N}^{-1} \mathbf{B} \mathbf{M}^{-1/2}) = \Gamma_L^2 / \gamma_L^2 \quad (27)$$

for the condition numbers with respect to the euclidean norm.

In Section 3 below we discuss the Kronecker structure of the system matrices $\mathbf{B}$ and $\hat{\mathbf{B}}$, the norm matrix $\mathbf{N}$, and the preconditioner $\mathbf{M}$. We then describe the structure of the inverses $\mathbf{N}^{-1}$ and $\mathbf{M}^{-1}$, and their practical approximations based on a wavelet-in-time multigrid-in-space transform.

2.5. Norm equivalence on the trial space. As noted above, the two norms $\|\cdot\|_X$ and $\|\cdot\|_X$, related by $\|w\|_X^2 = \|w\|_X^2 + \|w(T)\|_H^2$, are equivalent. Since the norm $\|\cdot\|_X$ gives rise to a simpler preconditioner than (26), it is of interest to quantify this norm equivalence. Obviously, $\|\cdot\|_X \leq \|\cdot\|_X$. For the reverse comparison we need to find the constant C_X in the estimate $\|w(T)\|_H \leq C_X \|w\|_X$, from which $\|w\|_X^2 \leq (1 + C_X^2) \|w\|_X^2$ follows. We shall do this only in the (relevant) situation that H is infinite-dimensional and the embedding $V \hookrightarrow H$ is compact. In this case there exists a countable orthonormal basis $\{\sigma_k\}_{k \in \mathbb{N}}$ for H , which is also an orthogonal basis for V and V' . Set $\lambda_k := \|\sigma_k\|_V$. The indexing is assumed to be such that $0 < \lambda_1 \leq \lambda_k$ for all $k \in \mathbb{N}$. Then $1/\lambda_1$ is the norm of the embedding $V \hookrightarrow H$, appearing in the Friedrichs inequality when V and H are Sobolev spaces. Further, by orthonormality of the basis in H there follows $\lambda_k^{-1} = \|\sigma_k\|_{V'}$. Expanding $w \in X$ into this basis we can write $w = \sum_{k \in \mathbb{N}} w_k \otimes \sigma_k$, which implies $\|w(0)\|_H^2 = \sum_{k \in \mathbb{N}} |w_k(0)|^2$, $\|w\|_{L_2(J;V)}^2 = \sum_{k \in \mathbb{N}} \|\lambda_k w_k\|_{L_2(J)}^2$ and $\|\partial_t w\|_{L_2(J;V')}^2 = \sum_{k \in \mathbb{N}} \|\lambda_k^{-1} w'_k\|_{L_2(J)}^2$.

This motivates the reduced question: Given $\lambda > 0$, find the smallest $C > 0$ such that for all $f \in H^1(J)$ there holds $C^{-2} |f(0)|^2 \leq \|\lambda f\|_{L_2(J)}^2 + \|\lambda^{-1} f'\|_{L_2(J)}^2$.

Herein, we can without loss of generality suppose that $f(0) = 1$. Then the question becomes that of minimizing the right hand side. Applying the variational principle leads to the boundary value problem $-\lambda^{-2}f'' + \lambda^2f = 0$ with the boundary conditions $f'(0) = 0$ and $f(T) = 1$. This problem can be solved explicitly, yielding the catenary as the solution. A direct computation then shows $C = 1/\sqrt{\tanh(\lambda^2 T)}$. This is a decreasing function of λ . Therefore, the best constant in $\|w(T)\| \leq C_X \|w\|_X$ is given by $C_X^2 := 1/\tanh(\lambda_1^2 T)$. As $T \rightarrow \infty$, we have $C_X \searrow 1$. For instance, if $\lambda_1^2 T \geq 1$ then $C_X \leq 1.1459$. On the other hand, if T is small, we can approximate $\tanh(\lambda_1^2 T) \sim \lambda_1^2 T$, so that $C_X \sim 1/\sqrt{\lambda_1^2 T}$ diverges as $T \searrow 0$.

In summary, if T is large enough, then the last term in (26) can be omitted without sacrificing the quality of the preconditioner $\mathbf{M}$.

3. Wavelet-in-time multigrid-in-space approximation. In order to simplify the further exposition of the main ideas relating to preconditioning, we shall assume from now on that in the parabolic evolution equation (4), the generator $t \mapsto A(t)$ is time-independent. Hence we write A instead of $A(t)$. While the results are stable with respect to slight perturbations of these assumptions, the discretization of the operators (8) and (12) becomes technical, and the quality of the proposed preconditioner may deteriorate for large variations of $t \mapsto A(t)$.

The assumption that A is time-independent allows us to equip the space V with the norm $\|\chi\|_V := (A\chi, \chi)^{1/2}$, which by (1)–(3) is equivalent to the original norm.

In order to avoid in practice the computation of the dual norm $\|\cdot\|_{V'}$ that appears in the definition (7) of the norm $\|\cdot\|_X$, for any nontrivial finite-dimensional subspace $V_L \subset V$ we introduce $\|\chi\|_{V'_L}$ as the norm of the functional $\langle \chi, \cdot \rangle \in V'$ restricted to V_L , and the measure of self-duality $0 < \kappa_L \leq 1$ as the largest constant satisfying

$$\|\chi\|_{V'_L} \leq \|\chi\|_{V'} \leq \kappa_L^{-1} \|\chi\|_{V'_L} \quad \forall \chi \in V_L. \quad (28)$$

Equivalently, κ_L^{-1} is the norm of the H -orthogonal projector from V to $V_L \subset V$, see [3, Lemma 6.2] and the discussion therein, as well as [10] for a quantitative analysis for finite element subspaces.

3.1. Kronecker product structure of discretized operators. Recall from (16) that the discrete trial and test spaces $X_L \subset X$ and $Y_L \subset Y$ are of the tensor product form $X_L = E_L \otimes V_L$ and $Y_L = (F_L \otimes V_L) \times V_L$. This has implications on the structure of the system matrices $\mathbf{B}$ and $\hat{\mathbf{B}}$ introduced in Section 2.2.

Let now $\Theta \subset E_L$, $\Xi \subset F_L$, and $\Sigma \subset V_L$, be bases for the respective spaces. Set $\Phi := \{\theta \otimes \sigma : (\theta, \sigma) \in \Theta \times \Sigma\}$ and $\Psi_1 := \{\xi \otimes \sigma : (\xi, \sigma) \in \Xi \times \Sigma\}$. Then

$$\Phi \subset X_L \quad \text{and} \quad \Psi := (\Sigma \times \{0\}) \cup (\{0\} \times \Psi_1) \subset Y_L \quad (29)$$

are bases for X_L and Y_L , respectively.

The system matrix $\mathbf{B} \in \mathbb{R}^{\Psi \times \Phi}$ from Section 2.2, whose components are $\mathbf{B}_{\psi\phi} = \langle B\phi, \psi \rangle$, then has the form

$$\mathbf{B} = \begin{pmatrix} \mathbf{C}_t^{FE} \otimes \mathbf{M}_x + \mathbf{M}_t^{FE} \otimes \mathbf{A}_x \\ \mathbf{e}_0 \otimes \mathbf{M}_x \end{pmatrix}, \quad (30)$$

where $\mathbf{a}$ “temporal FEM” matrices $\mathbf{C}_t^{FE}, \mathbf{M}_t^{FE} \in \mathbb{R}^{\Xi \times \Theta}$ and the row vector $\mathbf{e}_0 \in \mathbb{R}^{1 \times \Theta}$ have the components

$$[\mathbf{C}_t^{FE}]_{\xi\theta} = (\theta', \xi)_{L_2(J)}, \quad [\mathbf{M}_t^{FE}]_{\xi\theta} = (\theta, \xi)_{L_2(J)}, \quad [\mathbf{e}_0]_{1\theta} = \theta(s), \quad (31)$$

and **b)** the usual “spatial FEM” mass and stiffness matrices $\mathbf{M}_x, \mathbf{A}_x \in \mathbb{R}^{\Sigma \times \Sigma}$ have the components

$$[\mathbf{M}_x]_{\tilde{\sigma}\sigma} = (\tilde{\sigma}, \sigma), \quad [\mathbf{A}_x]_{\tilde{\sigma}\sigma} = (A\tilde{\sigma}, \sigma). \quad (32)$$

Let $\mathbf{M}_t^E$ and $\mathbf{M}_t^F$ denote the temporal mass matrices for $\Theta \subset E_L$ and $\Xi \subset F_L$, respectively, and $\mathbf{A}_t^E$ the temporal stiffness matrix for $\Theta \subset E_L$. Let $\hat{\mathbf{A}}_x$ be the symmetric part of $\mathbf{A}_x$. Then the preconditioner $\mathbf{M}$ from Section 2.4 satisfies

$$\mathbf{M} \sim \overline{\mathbf{M}} + (\mathbf{e}_T^\top \otimes \mathbf{e}_T) \otimes \mathbf{M}_x, \quad (33)$$

where

$$\overline{\mathbf{M}} := \mathbf{M}_t^E \otimes \hat{\mathbf{A}}_x + \mathbf{A}_t^E \otimes (\mathbf{M}_x \hat{\mathbf{A}}_x^{-1} \mathbf{M}_x). \quad (34)$$

Note that $\mathbf{e}_T^\top \otimes \mathbf{e}_T$ in (33) is a rank-one matrix in $\mathbb{R}^{\Theta \times \Theta}$ of the same size as $\mathbf{M}_t^E$ and $\mathbf{A}_t^E$. Employing the estimate (28) for the last term of (34), together with the observation $\|\Sigma^\top \mathbf{c}\|_{V_L'}^2 = \mathbf{c}^\top \mathbf{M}_x \hat{\mathbf{A}}_x^{-1} \mathbf{M}_x \mathbf{c}$, shows the equivalence in (33).

The norm-measuring matrix $\mathbf{N}$ from Section 2.3 is the block matrix

$$\mathbf{N} = \begin{pmatrix} \mathbf{M}_t^F \otimes \hat{\mathbf{A}}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_x \end{pmatrix}. \quad (35)$$

Finally, the matrix $\hat{\mathbf{B}}$ from Section 2.2 has the form

$$\hat{\mathbf{B}} = \mathbf{Z} + \mathbf{M}_t^E \otimes \hat{\mathbf{A}}_x + (\mathbf{e}_T^\top \otimes \mathbf{e}_T) \otimes \mathbf{M}_x, \quad (36)$$

where $\mathbf{Z} \in \mathbb{R}^{\Phi \times \Phi}$ is a symmetric matrix with the components $\mathbf{Z}_{\varphi\phi} := \langle \hat{A}^{-1} \tilde{C}\varphi, \tilde{C}\phi \rangle$. Details for a concrete example of $\mathbf{Z}$ are discussed in Section 4.1.3.

3.2. Multigrid-in-space. As indicated in Section 2.4, we require a practical way to apply the inverses of $\mathbf{N}$ and of the preconditioner $\mathbf{M}$.

3.2.1. Inverse of $\mathbf{N}$. From (35) we have the representation

$$\mathbf{N}^{-1} = \begin{pmatrix} (\mathbf{M}_t^F)^{-1} \otimes \hat{\mathbf{A}}_x^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_x^{-1} \end{pmatrix} \quad (37)$$

for the inverse of $\mathbf{N}$. Using an L_2 -orthogonal basis for the temporal test space $F_L \subset L_2(J)$, see Section 4.1.4, renders $\mathbf{M}_t^F$ a diagonal matrix, so that $\mathbf{N}^{-1}$ is block-diagonal. The inverse $\hat{\mathbf{A}}_x^{-1}$ can be replaced by a multigrid iteration.

3.2.2. Inverse of $\mathbf{M}$. Based on the discussion in Section 2.5 and the equivalence (33), instead of $\mathbf{M}$ we will consider the simpler matrix $\overline{\mathbf{M}}$ as the preconditioner. For the sake of readability, we omit the superscript in the notation for the “temporal FEM” matrices. Let $\mathbf{V}_t$ be a transformation matrix of the same size as $\mathbf{A}_t$ and $\mathbf{M}_t$ such that $\mathbf{V}_t^\top \mathbf{M}_t \mathbf{V}_t$ and $\mathbf{V}_t^\top \mathbf{A}_t \mathbf{V}_t$ are spectrally equivalent to some diagonal matrices $\mathbf{J}_t$ and $\mathbf{D}_t$, respectively, where the constants of the equivalence should be close to one. We refer to the ratio of those constants as the condition number of the transformation. The canonical choice of $\mathbf{V}_t$ is the matrix collecting in its columns the $\mathbf{M}_t$ -orthonormal eigenvectors of the generalized eigenvalue problem $\mathbf{A}_t \mathbf{v} = \lambda \mathbf{M}_t \mathbf{v}$, in which case $\mathbf{J}_t := \mathbf{V}_t^\top \mathbf{M}_t \mathbf{V}_t$ is the identity and $\mathbf{D}_t := \mathbf{V}_t^\top \mathbf{A}_t \mathbf{V}_t$ has the eigenvalues on the diagonal. In this case, the condition number of the transformation is one, but the drawback

is that $\mathbf{V}_t$ is a dense matrix. A sparse alternative $\mathbf{V}_t := \mathbf{T}_t^\top$ is given by the inverse wavelet(-like) transformation introduced in Section 3.3 below, and we take $\mathbf{J}_t$ as the diagonal of $\mathbf{V}_t^\top \mathbf{M}_t \mathbf{V}_t$ and $\mathbf{D}_t$ as the diagonal of $\mathbf{V}_t^\top \mathbf{A}_t \mathbf{V}_t$. The condition numbers are investigated in Section 3.3.2. We suppose now that $\mathbf{V}_t$ is such that

$$\mathbf{V}_t^\top \mathbf{M}_t \mathbf{V}_t \sim \mathbf{J}_t \quad \text{and} \quad \mathbf{V}_t^\top \mathbf{A}_t \mathbf{V}_t \sim \mathbf{D}_t \quad (38)$$

with some nonnegative diagonal matrices $\mathbf{J}_t$ and $\mathbf{D}_t$, where $\mathbf{J}_t$ is positive definite (specifically, we use (48)). Let $\mathbf{I}_x$ denote the identity matrix of the same size as $\mathbf{M}_x$ and $\mathbf{A}_x$. We set $\mathbf{T} := \mathbf{V}_t^\top \otimes \mathbf{I}_x$. Then $\mathbf{T}^\top = \mathbf{V}_t \otimes \mathbf{I}_x$.

To obtain a computationally accessible approximation of $\mathbf{M}^{-1}$, we begin with the change of temporal basis

$$\mathbf{M}^{-1} \sim \overline{\mathbf{M}}^{-1} = \mathbf{T}^\top (\mathbf{T} \overline{\mathbf{M}} \mathbf{T}^\top)^{-1} \mathbf{T} \sim \mathbf{T}^\top (\mathbf{J}_t \otimes \widehat{\mathbf{A}}_x + \mathbf{D}_t \otimes (\mathbf{M}_x \widehat{\mathbf{A}}_x^{-1} \mathbf{M}_x))^{-1} \mathbf{T}. \quad (39)$$

This requires the approximation of the inverse of the block-diagonal matrix

$$\mathbf{J}_t \otimes \widehat{\mathbf{A}}_x + \mathbf{D}_t \otimes (\mathbf{M}_x \widehat{\mathbf{A}}_x^{-1} \mathbf{M}_x), \quad (40)$$

where each block has the form $j^2 \widehat{\mathbf{A}}_x + d^2 (\mathbf{M}_x \widehat{\mathbf{A}}_x^{-1} \mathbf{M}_x)$ with some nonnegative reals j and d , and we recall that $\widehat{\mathbf{A}}_x$ denotes the symmetric part of the spatial FEM stiffness matrix $\mathbf{A}_x$. Thus, the inversion of the space-time matrix (40) is equivalent to a sequence of *independent* spatial problems of the form

$$j^2 \mathbf{H}_x^\gamma \mathbf{p} = \mathbf{q} \quad \text{with} \quad \mathbf{H}_x^\gamma := \widehat{\mathbf{A}}_x + \gamma^2 (\mathbf{M}_x \widehat{\mathbf{A}}_x^{-1} \mathbf{M}_x), \quad (41)$$

where $\gamma = d/j$. Intuitively, these spatial blocks measure the energy content of the part of the solution associated to the temporal basis function of frequency γ . Now, the identity $\mathbf{H}_x^\gamma = (\widehat{\mathbf{A}}_x + i\gamma \mathbf{M}_x) \widehat{\mathbf{A}}_x^{-1} (\widehat{\mathbf{A}}_x - i\gamma \mathbf{M}_x)$, where i is the imaginary unit, is easily verified. Thus, solving (41) is equivalent to solving two Helmholtz problems with the imaginary parameter $\pm i\gamma$, with one forward application of $\widehat{\mathbf{A}}_x$ between them. In order to avoid complex numbers arithmetics, one can observe that the right-hand side of the identity is the Schur complement of a 2×2 block matrix that admits iterative solvers that are robust in γ [54]. Recalling, however, that $\mathbf{M}$ is merely a preconditioner, we estimate $\mathbf{H}_x^\gamma$ by the composition of two Helmholtz problems

$$\mathbf{H}_x^\gamma \leq \overline{\mathbf{H}}_x^\gamma := (\widehat{\mathbf{A}}_x + \gamma \mathbf{M}_x) \widehat{\mathbf{A}}_x^{-1} (\widehat{\mathbf{A}}_x + \gamma \mathbf{M}_x) \leq 2\mathbf{H}_x^\gamma \quad (42)$$

with a real frequency γ . Indeed, one only has to verify $2\gamma \mathbf{M}_x \leq \mathbf{H}_x^\gamma$, but this follows by changing to the eigenvalue basis of the generalized eigenvalue problem $\widehat{\mathbf{A}}_x \mathbf{p} = \mu \mathbf{M}_x \mathbf{p}$ together with the estimate $2\gamma \leq \mu + \gamma^2/\mu$. Therefore, in the inverse of (40), which is again block-diagonal, we replace each spatial block by the corresponding real double Helmholtz problem, that is, by the application of the mapping

$$\mathbf{q} \mapsto j^{-2} (\widehat{\mathbf{A}}_x + \gamma \mathbf{M}_x)^{-1} \widehat{\mathbf{A}}_x (\widehat{\mathbf{A}}_x + \gamma \mathbf{M}_x)^{-1} \mathbf{q}. \quad (43)$$

The exact solution of these positive definite Helmholtz problems can be replaced by one of the following strategies:

1. The multigrid iteration, which is robust in $\gamma > 0$ [43].
2. The (incomplete) LU or Cholesky factorization of the matrix $\widehat{\mathbf{A}}_x + \gamma \mathbf{M}_x$. This is simpler to implement than the multigrid iteration but may not be applicable to very fine spatial discretizations.

3. The preconditioner from [14, Section 4.2], again robust in γ . In fact, it easily extends to a preconditioner for the compound matrix $\bar{\mathbf{H}}_x^\gamma$ in (42) and can therefore replace the mapping (43).
4. A wavelet transform and a diagonal matrix like in the temporal direction (38), which is robust in γ if the wavelets form Riesz bases for V and H .

As a further simplification, γ may be rounded to one of reference values $\{\gamma_i^{\text{ref}}\}$, for instance logarithmically equispaced such that for any occurring γ there is a reference value γ_i^{ref} with $|\log_2(\gamma/\gamma_i^{\text{ref}})| \leq 1/2$. This approximation allows to speed up the application of the preconditioner through the recycling of matrices in the 1st strategy or factorizations in the 2nd strategy, at the cost of another factor of two in the spectral equivalence (39) and additional memory consumption. In the numerical experiments we will use the 1st and the 2nd strategy with $\gamma_i^{\text{ref}} := 2^i$. In view of parallelization we reiterate here that (40) is a block-diagonal matrix.

In summary, the following approximations have been made to obtain a tractable version of the inverse of $\mathbf{M}$: replace the $\|\cdot\|_V$ norm by the $\|\cdot\|_{V_L'}$ norm using (28); omit the final-time term in (33) as quantified in Section 2.5; block-diagonalize with (38); round the frequency γ to one of the reference values; approximate the inverse of each block through one of the four strategies above.

3.3. Construction of the inverse wavelet transform. In this section we construct the inverse wavelet transform $\mathbf{T}_t$ such that the sparse matrix $\mathbf{V}_t := \mathbf{T}_t^\top$ has the properties required in Section 3.2.2. The main idea is to identify the “most energetic” hat functions that lie in the top energy bandwidth of relative width $\eta > 0$, approximately orthogonalize them to the remaining ones thus defining the fine scale subspace, and to proceed recursively. A similar construction was given and analyzed in [51, 52] but on a predefined hierarchy of meshes.

3.3.1. Construction. Let $\bar{\mathcal{T}}$ denote the collection of finite temporal meshes $\mathcal{T} = \{0 =: t_0 < t_1 < \dots < t_N := T\}$. Recall that $J = (0, T)$. For any $\mathcal{T} \in \bar{\mathcal{T}}$ let $\Theta(\mathcal{T})$ denote the set of hat functions (hats) $\mathcal{T}$, that is the set of piecewise linear splines θ_t on $\mathcal{T}$ that evaluate to one on some $t \in \mathcal{T}$ and to zero on all other nodes. We let $s(\mathcal{T})$ denote the left-most node $s \in \mathcal{T} \setminus \{0, T\}$ for which the energy $\epsilon(\mathcal{T}) := \|\theta_s\|_{H^1(J)}^2 / \|\theta_s\|_{L_2(J)}^2$ is maximized among the non-boundary hats $\theta_t \in \Theta(\mathcal{T}) \setminus \{\theta_0, \theta_T\}$.

A wavelet-like basis for the span of $\Theta(\mathcal{T})$ is constructed as follows.

Assume given:

1. $\mathcal{T} \in \bar{\mathcal{T}}$ with $\#\mathcal{T} \geq 3$.
2. Relative energy bandwidth parameter $\eta > 1$ (we will use $\eta := 1.9$).
3. Number of orthogonalization steps $\nu \in \mathbb{N}$.

Step 0. Set $\Theta_0(\mathcal{T}) := \Theta(\mathcal{T})$ and $K := 0$.

Step 1. Given $\mathcal{T} \in \bar{\mathcal{T}}$, identify fine $\mathcal{T}_+ \subset \mathcal{T}$ and coarse nodes $\mathcal{T}_- \subset \mathcal{T}$ by:

1. Set $\mathcal{T}_0 := \mathcal{T}$ and $n := 0$.
2. While $\#\mathcal{T}_n \geq 3$ and $\epsilon(\mathcal{T}_n) \geq \eta^{-1}\epsilon(\mathcal{T}_0)$ do:
Set $\mathcal{T}_{n+1} := \mathcal{T}_n \setminus s(\mathcal{T}_n)$ and increase n by 1.
3. Define $\mathcal{T}_- := \mathcal{T}_n$ and $\mathcal{T}_+ := \mathcal{T} \setminus \mathcal{T}_-$.

Please note that, after the n -th iteration, $\Theta(\mathcal{T}_{n+1})$ does not equal $\Theta(\mathcal{T}_n) \setminus \{\theta_{s(\mathcal{T}_n)}\}$ because removing a node from $\mathcal{T}_n$ modifies the hats associated with the two neighboring nodes of $s(\mathcal{T}_n) \in \mathcal{T}_n$ by widening their support. Yet, obviously $\Theta(\mathcal{T}_n)$ and $\{\theta_{s(\mathcal{T}_n)}\} \cup \Theta(\mathcal{T}_{n+1})$ span the same space. On a uniform grid $\mathcal{T}$, in particular, the algorithm collects every second node of $\mathcal{T} \setminus \{0, T\}$ in $\mathcal{T}_-$ and every other in $\mathcal{T}_+$. Also, each intermediate $\Theta(\mathcal{T}_n)$ constitutes a partition of unity on the interval J , and so do

the coarse hats $\Theta(\mathcal{T}_-)$.

Step 2. Let $\Theta_+(\mathcal{T})$ denote the set of hats that are removed from $\Theta(\mathcal{T}_n)$ over all iterations n during Step 1. In general, the set of obtained fine hats $\Theta_+(\mathcal{T})$ is not the same as $\{\theta_t \in \Theta(\mathcal{T}) : t \in \mathcal{T}_+\}$ because neighboring nodes of $\mathcal{T}$ may have been selected into $\mathcal{T}_+$. Now, the basis $\Theta_+(\mathcal{T}) \cup \Theta(\mathcal{T}_-)$ is a two-scale basis for the span of $\Theta(\mathcal{T})$. To obtain a wavelet-like basis $\Psi_+(\mathcal{T})$ for the detail space spanned by $\Theta_+(\mathcal{T})$ we perform an approximate orthogonalization of the fine hats $\Theta_+(\mathcal{T})$ against the coarse hats $\Theta(\mathcal{T}_-)$ using the mapping

$$P_{\mathcal{T}_-} : L_2(J) \rightarrow L_2(J), \quad \theta \mapsto P_{\mathcal{T}_-} \theta := \theta - \sum_{\zeta \in \Theta(\mathcal{T}_-)} \frac{(\zeta, \theta)_{L_2(J)}}{(\zeta, 1)_{L_2(J)}} \zeta, \quad (44)$$

which can be interpreted as a Jacobi preconditioned Richardson iteration for the exact L_2 orthogonalization. The wavelet-like basis $\Psi_+(\mathcal{T})$ is defined by

$$\Psi_+(\mathcal{T}) := \{[P_{\mathcal{T}_-}]^\nu \theta : \theta \in \Theta_+(\mathcal{T})\}, \quad (45)$$

where $\nu \in \mathbb{N}$ was given. An application of $P_{\mathcal{T}_-}$ to a compactly supported θ increases its support by at most the support of the few coarse hats that are nonzero there.

Step 3. Call $\Psi_K := \Psi_+(\mathcal{T})$ and $\Theta_{K+1} := \Theta(\mathcal{T}_-)$. If $\#\mathcal{T}_- = 2$ then terminate. If $\#\mathcal{T}_- \geq 3$, increase K by one, and continue with Step 1 for $\mathcal{T} := \mathcal{T}_-$.

Once the algorithm terminates, we have constructed a multiscale basis

$$\Psi(\mathcal{T}) := \Theta_{K+1} \cup \Psi_K \cup \dots \cup \Psi_1 \cup \Psi_0 \quad (46)$$

for the span of the original hats $\Theta(\mathcal{T})$. Since the number of levels is not known in advance, $K+1$ denotes the coarsest while 0 denotes the finest level.

Clearly, the mapping $P_{\mathcal{T}_-}$ in (44) operates only in the subspace spanned by $\Theta(\mathcal{T}_-)$ leaving its orthogonal complement in $L_2(J)$ unchanged. Since $\Theta(\mathcal{T}_-)$ constitute a partition of unity on J we have $P_{\mathcal{T}_-} 1 = 1 - \sum_{\zeta \in \Theta(\mathcal{T}_-)} \zeta = 0$. Therefore, $P_{\mathcal{T}_-} \theta$ has at least one vanishing moment, so it is indeed a wavelet-like oscillating function. In the limit $\nu \rightarrow \infty$, the function $[P_{\mathcal{T}_-}]^\nu \theta$ is orthogonal to the span of $\Theta(\mathcal{T}_-)$.

Some typical wavelet-like basis functions are shown in Figure 3. On the uniform mesh, there is a striking similarity to the $_{2,2\nu}\psi$ family of wavelets of [19]. But unlike those, our wavelet-like functions do not have 2ν vanishing moments, hence are not exactly the same.

In the Appendix we provide a Matlab code that constructs the coefficient transformation matrix $\mathbf{T}_t$ such that the mass matrix with respect to the multiscale basis is $\mathbf{T}_t \mathbf{M}_t \mathbf{T}_t^\top$ where $\mathbf{M}_t$ is the mass matrix with respect to the original hat basis $\Theta(\mathcal{T})$. The mapping $\mathbf{w} \mapsto \mathbf{T}_t^\top \mathbf{w}$ is the inverse wavelet transformation, as it takes a vector of coefficients $\mathbf{w}$ with respect to the multiscale wavelet-like basis $\Psi(\mathcal{T})$ and returns a vector of coefficients with respect to the hat basis $\Theta(\mathcal{T})$.

The transformation can be performed in a matrix-free fashion with linear complexity known from wavelet pyramid schemes. However, we are thinking of the situation where it is applied (from the right) to a distributed matrix that stores spatial vectors in its columns. This is because $(\mathbf{T}_t \otimes \mathbf{I}_x) \text{Vec}(\mathbf{U}) = \text{Vec}(\mathbf{U} \mathbf{T}_t^\top)$ is precisely the type of operation required in (39), where $\mathbf{U}$ is a matrix of suitable size and $\text{Vec}(\cdot)$ stacks the columns of its argument one after the other into a long vector; see Section 4.2.4. We therefore believe that it may be beneficial, at least at first, to formulate and test the transformation in matrix form.

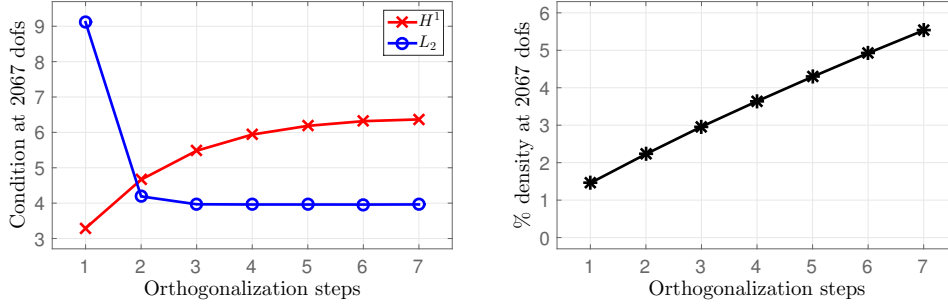


FIG. 1. Condition number of the wavelet-like basis (left) and percentage of nonzeros in the transformation matrix (right) as a function of the number of orthogonalization steps, see Section 3.3.2. Without orthogonalization, the condition number in L_2 is very large.

3.3.2. Condition numbers of the basis. We compute the condition number of the wavelet-like basis $\Psi(\mathcal{T}^k)$ constructed in Section 3.3.1 for a test sequence of temporal meshes $\mathcal{T}^k$, $k = 1, \dots, 11$. Each $\mathcal{T}^k$ contains a uniform temporal mesh on $[0, T]$ with $2^k + 1$ nodes and additionally the nodes $1/2 \pm 2^{-r}$, $r = 1, \dots, 20$, that provide geometric refinement towards $t = 1/2$. The condition number in $L_2(J)$ and $H^1(J)$ is the ratio of the Riesz basis constants of the normalized wavelet-like basis in $L_2(J)$ and in $H^1(J)$, respectively. These constants are precisely the extremal eigenvalues of $\{\mathbf{T}_t \mathbf{M}_t \mathbf{T}_t^\top\}$ and $\{\mathbf{T}_t \mathbf{A}_t \mathbf{T}_t^\top\}$, where $\{\mathbf{X}\}$ denotes the rescaled matrix

$$\{\mathbf{X}\} := (\text{diag} \mathbf{X})^{-1/2} \mathbf{X} (\text{diag} \mathbf{X})^{-1/2}, \quad (47)$$

and $\mathbf{M}_t$ and $\mathbf{A}_t$ are the mass and the stiffness matrix with respect to the original hat basis $\Theta(\mathcal{T}^k)$, cf. Section 3.3. Setting

$$\mathbf{J}_t := \text{diag}(\mathbf{V}_t^\top \mathbf{M}_t \mathbf{V}_t) \quad \text{and} \quad \mathbf{D}_t := \text{diag}(\mathbf{V}_t^\top \mathbf{A}_t \mathbf{V}_t) \quad (48)$$

for $\mathbf{V}_t := \mathbf{T}_t^\top$, equivalence (38) is satisfied with those constants. Additionally, we measure the percentage of nonzeros in the transformation matrix $\mathbf{T}_t$.

The results depending on the number ν of orthogonalization steps are shown in Figure 1 for $\mathcal{T}^k$ with $k = 11$. It seems that $\nu = 2$ is the appropriate choice because for larger ν the condition number in $L_2(J)$ remains stable while that in $H^1(J)$ increases slightly. Moreover, the percentage of nonzeros increases roughly linearly with ν , as long as ν is small.

The dependence of the condition number on the number of nodes in $\mathcal{T}^k$, $k = 1, \dots, 11$, is shown in Figure 2 for $\nu = 1, 2, 3$, orthogonalization steps. The condition number in $L_2(J)$ is robust as the number of nodes increases, and that in $H^1(J)$ exhibits a slight growth. Again, $\nu = 2$ seems to be a good choice.

4. Numerical results. In Section 4.1 we document the quality of the space-time preconditioners using a series of numerical examples in one spatial dimension. In Section 4.2 we apply the preconditioner to the space-time resolution of a model parabolic evolution equation in two spatial dimensions with parallelization in time.

4.1. Quality of the space-time preconditioner. In this section we investigate the quality of the proposed preconditioner on small scale examples by computing the condition numbers of the preconditioned system.

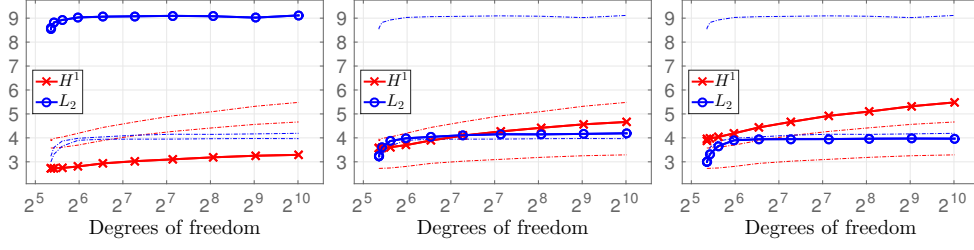


FIG. 2. Condition number of the wavelet-like basis as a function of the size of the test mesh for $\nu = 1, 2, 3$ (left to right) orthogonalization steps.

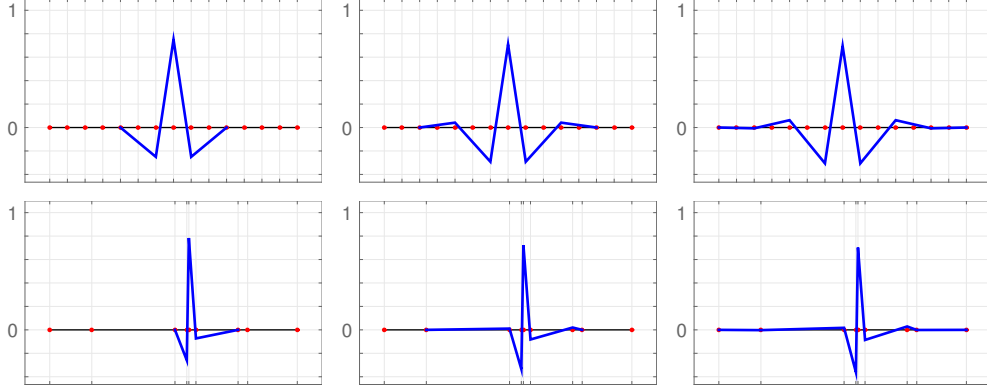


FIG. 3. Example of a wavelet-like basis function for $\nu = 1, 2, 3$ (left to right). Top: uniform mesh, bottom: highly non uniform mesh. The mesh is indicated by the vertical lines and the dots on the horizontal axis. Note the increasing support with ν .

4.1.1. Model problem. In order to investigate the performance of the space-time preconditioners in detail, we look at the model problem of convection-diffusion on the interval $D := (0, 1)$. We fix the end-time $T := 1$. Specifically, we assume for (1)–(3) that $H := L_2(D)$, $V := H_0^1(D)$, and

$$A := -\partial_x^2 + \beta \partial_x \quad (49)$$

with some constant $\beta \in \mathbb{R}$. The symmetric part of A is then $\hat{A} = -\partial_x^2$, and the asymmetric part is $\tilde{A} = \beta \partial_x$. Of particular interest is the performance of the preconditioner with respect to the drift velocity β .

We recall from the introduction of Section 3 that the norm on $V = H_0^1(D)$ is taken as $\|\chi\|_V := \|\partial_x \chi\|_{L_2(D)}$. Then inequality (3) holds with $\alpha = 1$ and $\gamma_0 = 0$.

4.1.2. Computation of the condition numbers. The condition number κ_2 of the preconditioned systems (27) with different approximate variants of the inverses is computed by means of a power iteration for the maximal and minimal eigenvalue of the associated generalized eigenvalue problems. The condition number is then obtained as the ratio of the maximal to the minimal eigenvalue. For instance, in order to approximate the maximal eigenvalue of $\mathbf{M}^{-1}\hat{\mathbf{B}}$ we iterate $\tilde{\mathbf{x}}_n := \mathbf{M}^{-1}\hat{\mathbf{B}}\mathbf{x}_{n-1}$, $e_n^{\max} := |\tilde{\mathbf{x}}_n|$, $\mathbf{x}_n = \tilde{\mathbf{x}}_n / e_n^{\max}$, starting with an all-ones vector. Then $e_n^{\max}$ converges to the maximum eigenvalue, and we stop the iteration when the relative improvement is less than 10^{-4} . In order to compute the minimal eigenvalue, more work is required.

Here, we iterate $\tilde{\mathbf{x}}_n := \mathbf{M}\hat{\mathbf{B}}^{-1}\mathbf{x}_{n-1}$, $e_n^{\min} := 1/|\tilde{\mathbf{x}}_n|$, $\mathbf{x}_n := \tilde{\mathbf{x}}_n e_n^{\min}$, starting with an all-ones vector, upon which $e_n^{\min}$ converges to the minimum eigenvalue. The application of the inverse of $\hat{\mathbf{B}}$ is replaced by a preconditioned conjugate gradient iteration with preconditioner $\mathbf{M}^{-1}$ and tolerance 10^{-6} . We point out that in the case of multigrid approximation of the blocks of $\mathbf{M}^{-1}$, say $\mathbf{W} \approx \mathbf{M}^{-1}$, one needs the inverse $\mathbf{W}^{-1} \approx \mathbf{M}$ in the inverse power iteration. To that end we compute the matrix representation of the multigrid iteration and subsequently its inverse. We use a symmetric version of the multigrid where applicable. The overall computational effort to determine the condition numbers limits our numerical experiments to one-dimensional spatial domains. All computations are done in MATLAB R2014b.

4.1.3. Symmetric space-time variational formulation. For the symmetric space-time variational formulation (14) we assume in the numerical experiments that the spatial subspace $V_L \subset V$ are given as the L -dimensional space of polynomials of degree at most $(L+1)$ satisfying the boundary conditions of $H_0^1(D)$. A suitable basis $\Sigma \subset V_L$ is given by Babuška–Shen basis, which is the set of integrated Legendre polynomials $IP_k : t \mapsto \int_0^t P_k(s)ds$, $k = 1, \dots, L$, where P_k is the shifted Legendre polynomial on the interval $D = (0, 1)$ of degree k normalized in $L_2(D)$. The temporal discretization is achieved by taking $E_L \subset H^1(J)$ as the space of continuous piecewise affine functions on J with respect to a uniform partition of J . The number of the intervals will be specified below. This then defines the tensor product trial and test space X_L by (16).

Recall from Section 3.1 that the space-time system matrix $\hat{\mathbf{B}}$ contains the matrix $\mathbf{Z}$. It has the components $\mathbf{Z}_{\varphi\phi} = \langle \hat{A}^{-1}\tilde{C}\varphi, \tilde{C}\phi \rangle$, where $\varphi, \phi \in \Phi$ are the space-time tensor product basis functions for X_L from (29), and $\tilde{C} = d_t + \hat{A}$. The computation of this matrix is delicate due to the inverse of $\hat{A}$. Our choice of V_L is motivated by the fact that the action of the asymmetric part $\hat{A}$ and of the inverse $\hat{A}^{-1}$ on functions in V_L can be computed exactly; thus we can compute the spatial matrices $[\mathbf{A}_x^{11}]_{\tilde{\sigma}\sigma} := (\hat{A}^{-1}\tilde{A}\tilde{\sigma}, \tilde{A}\sigma)$, $[\mathbf{A}_x^{01}]_{\tilde{\sigma}\sigma} := (\hat{A}^{-1}\tilde{\sigma}, \tilde{A}\sigma)$, $[\mathbf{A}_x^{00}]_{\tilde{\sigma}\sigma} := (\hat{A}^{-1}\tilde{\sigma}, \sigma)$, and $\mathbf{A}_x^{10} := (\mathbf{A}_x^{01})^\top$, where as before, the basis $\tilde{\sigma}, \sigma \in \Sigma$ is used to index the components. In addition to the temporal FEM matrices from Section 3.1 we need the temporal advection matrix $[\mathbf{C}_t^{01}]_{\tilde{\theta},\theta} := (\tilde{\theta}, \theta')_{L_2(J)}$, where $\tilde{\theta}, \theta$ are elements of the temporal basis Θ , as well as its transpose $\mathbf{C}_t^{10} := (\mathbf{C}_t^{01})^\top$. The matrix $\mathbf{Z}$ can now be written as

$$\mathbf{Z} = (\mathbf{A}_t^E \otimes \mathbf{A}_x^{00}) + (\mathbf{C}_t^{01} \otimes \mathbf{A}_x^{10}) + (\mathbf{C}_t^{10} \otimes \mathbf{A}_x^{01}) + (\mathbf{M}_t^E \otimes \mathbf{A}_x^{11}). \quad (50)$$

We compute the condition number $\kappa_2(\mathbf{M}^{-1/2}\hat{\mathbf{B}}\mathbf{M}^{-1/2}) = \hat{\Gamma}_L/\hat{\gamma}_L$ by the power iteration from Section 4.1.2, approximating the inverse $\mathbf{M}^{-1}$ as described in Section 3.2.2. For the required temporal transformation $\mathbf{V}_t$ we use the inverse wavelet transform from Section 3.3 with $\nu = 2$ orthogonalization steps. The subspace $V_L \subset V$ is spanned by $L = 20$ integrated Legendre polynomials (the results do not essentially depend on L). All spatial inverses, such as in (43) and (50) are computed exactly. For the values $\beta = 0, 10, 20, \dots, 100$, of the drift velocity in (49), the condition number of the preconditioned system matrix as a function of $N = 2^0, 2^1, 2^2, \dots, 2^{11}$, equidistant temporal intervals is shown in Figure 4 (left). For each value of β , the condition number increases with N up to a certain value, which seems to scale like β^2 . For $\beta \lesssim 10$ the condition number remains very small as expected, but deteriorates for larger β especially with increasing number of temporal intervals. In Figure 4 (right) we show the condition number where, motivated by the structure of $\hat{\mathbf{B}}$, we have added the β -dependent term $(\mathbf{M}_t^E \otimes \mathbf{A}_x^{11})$ to each block of the preconditioner (40); each block

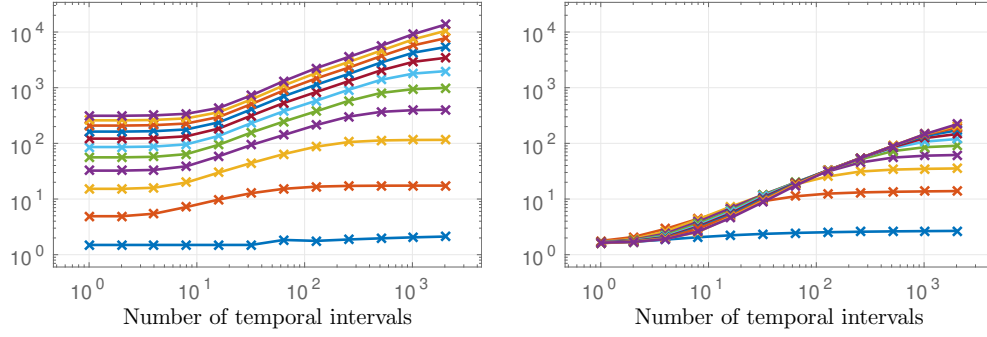


FIG. 4. Condition number of the preconditioned system matrix for the symmetric space-time variational formulation as a function of the number of temporal intervals, see Section 4.1.3. Each line from bottom to top corresponds to a value of drift velocity $\beta = 0, 10, \dots, 100$. Left: Wavelet-in-time and exact computation in space. Right: Same preconditioner with an additional β -dependent term.

is inverted exactly. This results in significantly lower condition numbers, which are however, still not robust in β .

Optimization problems constrained by evolution equations like (4) provide a major motivation for space-time simultaneous discretizations due to the coupling of the forward-in-time state equation and the backward-in-time adjoint equation, see for instance [45]. Consider the minimization of the standard tracking functional

$$\mathcal{J}(y, u) := \frac{1}{2} \int_J \|y(t) - y_\star(t)\|_{L_2(D)}^2 dt + \frac{\lambda}{2} \int_J \|u(t)\|_{L_2(D)}^2 dt, \quad (51)$$

where u is the control variable, $y_\star$ is the desired state, y is the state variable constrained by the parabolic evolution equation

$$y(0) = 0, \quad d_t y(t) + A y(t) = f(t) + u(t) \quad (\text{a.e.}) \quad t \in J, \quad (52)$$

and $\lambda > 0$ is a regularization parameter. Here A is given by (49) with zero drift $\beta = 0$. Introducing the dual variable p for the constraint, forming the first order optimality conditions, eliminating the control variable and discretizing the resulting linear system leads to the saddle-point problem

$$\mathbf{A} \begin{pmatrix} \mathbf{y} \\ \mathbf{p} \end{pmatrix} = \mathbf{rhs} \quad \text{where} \quad \mathbf{A} := \begin{pmatrix} \mathbf{M}_0 & \widehat{\mathbf{B}} \\ \widehat{\mathbf{B}} & -\lambda^{-1} \mathbf{M}_0 \end{pmatrix}, \quad (53)$$

for some vector $\mathbf{rhs}$ which is of no importance here. Here, $\mathbf{M}_0 := \mathbf{M}_t^E \otimes \mathbf{M}_x$ is the matrix corresponding to the space-time L_2 inner product. The large block matrix $\mathbf{A}$ is symmetric but indefinite, and therefore the system is typically solved using the preconditioned MINRES method [12, 53]. It is of particular interest to obtain computationally accessible preconditioners for (53) that are robust simultaneously in the discretization parameters and in the regularization parameter $\lambda > 0$. Having used the symmetric space-time variational formulation, we are in the situation that each block of the matrix is *symmetric*, so that the block-diagonal matrix with blocks $\mathbf{M}_0 + \sqrt{\lambda} \widehat{\mathbf{B}}$ and $\lambda^{-1}(\mathbf{M}_0 + \sqrt{\lambda} \widehat{\mathbf{B}})$ is a good preconditioner for $\mathbf{A}$, see [54, Section 4.1]. Replacing $\widehat{\mathbf{B}}$ by the spectrally equivalent $\mathbf{M}$ we obtain the block preconditioner

$$\mathbf{P} := \begin{pmatrix} \mathbf{M}_0 + \sqrt{\lambda} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \lambda^{-1}(\mathbf{M}_0 + \sqrt{\lambda} \mathbf{M}) \end{pmatrix}. \quad (54)$$

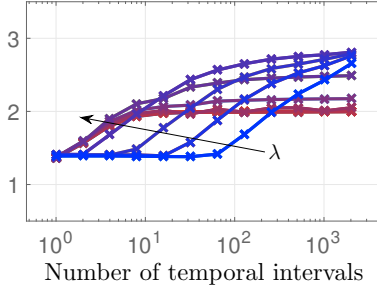


FIG. 5. Condition number of the preconditioned optimality system as a function of the number of temporal intervals. Each line corresponds to one of the values $\lambda = 10^{-5}, 10^{-4}, \dots, 10^5$ of the regularization parameter, as indicated by the arrow. Details in Section 4.1.3.

As in Section 3.2.2 we transform this preconditioner to a block-diagonal one which involves a series of independent spatial systems of the form $(\mathbf{M}_x + \sqrt{\lambda}\mathbf{H}_x^\gamma)\mathbf{p} = \mathbf{q}$ with $\mathbf{H}_x^\gamma$ from (41). These systems are small in our case and we solve them directly (but a decomposition analogous to (42) is possible). The resulting condition numbers $\kappa_2(\mathbf{P}^{-1/2}\mathbf{A}\mathbf{P}^{-1/2})$ with this approximation of $\mathbf{P}^{-1}$ are shown in Figure 5 for a range of N and λ . The condition number stays bounded by *three*.

4.1.4. Nonsymmetric space-time variational formulation. The nonsymmetric space-time variational formulation (10) is discretized using space-time tensor product subspaces (16) as described in Section 2.3. As $E_L \subset H^1(J)$ we take the space of continuous piecewise affine functions on J with respect to a uniform partition, as specified below. The test space $F_L \subset L_2(J)$ is taken as the space of piecewise constants with respect to a single uniform refinement of the same partition. The dimension of F_L is then roughly twice that of E_L . With the basis for F_L consisting of piecewise constant functions nonzero on exactly one of the subintervals, the matrix $\mathbf{N}$ defined in (35) and its inverse (37) are block-diagonal. The spatial discretization $V_L \subset H_0^1(D)$ is also given by continuous piecewise affine functions with respect to a uniform partition, the dimension will be specified below. For symmetric A , the resulting full tensor product discretization (16) satisfies the inf-sup condition (17) and the boundedness condition (18) with constants γ_L and Γ_L that depend only on the constants in the (1)–(3). In particular they can be estimated independently of the temporal and spatial resolution.

We compute the condition number $\kappa_2(\mathbf{M}^{-1/2}\mathbf{B}^\top\mathbf{N}^{-1}\mathbf{B}\mathbf{M}^{-1/2}) = \Gamma_L^2/\gamma_L^2$ by the power iteration as in Section 4.1.2, approximating the inverse $\mathbf{M}^{-1}$ as described in Section 3.2.2. The temporal transformation $\mathbf{V}_t$ is given by the inverse wavelet transform from Section 3.3 with $\nu = 2$ orthogonalization steps. Since the main difficulty is in the application of the inverse of $\mathbf{M}$, we apply the block-diagonal $\mathbf{N}^{-1}$ by direct inversion in each case. The spatial discretization $V_L \subset H_0^1(D)$ is fixed as the space of continuous piecewise affine functions with respect to the uniform partition of the interval D into 2^9 equal elements.

First, no multigrid approximation is employed, and all spatial matrices are inverted directly. The number of temporal elements is varied as $N = 2^0, 2^1, 2^2, \dots, 2^{11}$. The dependence of the condition number on the number of temporal elements N for different the drift velocities $\beta = 0, 0.01, \dots, 0.1$, is documented in Figure 6 (left). The condition number increases dramatically with N even for these small values of the drift velocity. Note that the drift velocity is here two orders of magnitude smaller than in Section 4.1.3.

Now we set $\beta = 0$ for the drift velocity and switch on the multigrid-in-space approximation in the application of the inverse of $\mathbf{M}$. We use the multigrid following

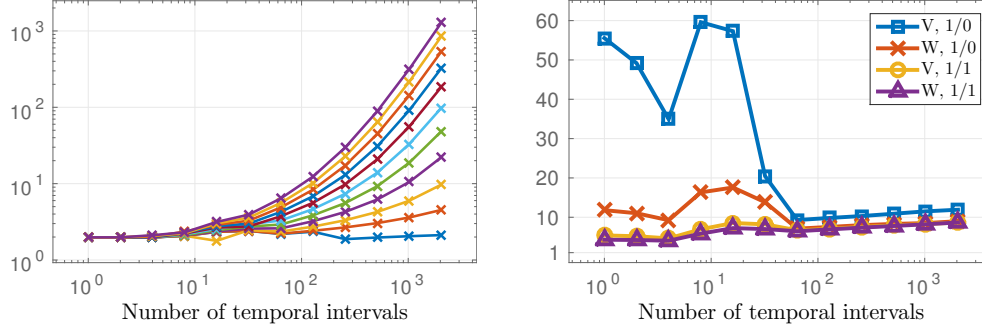


FIG. 6. Condition number of the preconditioned system matrix for the nonsymmetric space-time variational formulation as a function of the number of temporal intervals, see Section 4.1.4. Left: Wavelet-in-time and exact computation in space. Each line from bottom to top corresponds to a value of drift velocity $\beta = 0, 0.01, \dots, 0.1$. Right: For $\beta = 0$, wavelet-in-time multigrid-in-space preconditioner with different multigrid parameters: V- or W-cycle and the number of pre-/post-smoothing steps.

[30, Section 4.1]. The pre-smoother is defined by the Gauss-Seidel iteration $\mathbf{q} \mapsto (\mathbf{L} + \mathbf{D})^{-1}(-\mathbf{U}\mathbf{q} + \mathbf{f})$ and the post-smoother by $\mathbf{q} \mapsto (\mathbf{U} + \mathbf{D})^{-1}(-\mathbf{L}\mathbf{q} + \mathbf{f})$ for the equation $(\mathbf{L} + \mathbf{D} + \mathbf{U})\mathbf{q} = \mathbf{f}$ with strictly lower/upper triangular $\mathbf{L}/\mathbf{U}$ and diagonal $\mathbf{D}$. This ensures symmetry of the multigrid procedure when the number of pre- and post-smoothing steps is the same. As the prolongation we use the canonical embedding of V_k into V_{k+1} , the restriction is its transpose. In order to approximate the action of (43), the multigrid procedure with one pre-smoothing step is applied with the matrix $\hat{\mathbf{A}}_x + \gamma\mathbf{M}_x$, then $\hat{\mathbf{A}}_x$ is applied, then again the multigrid procedure. For four configurations of the multigrid determined by whether the V- or the W-cycle is executed, and whether none or one post-smoothing step is performed, the resulting condition numbers for the space-time preconditioned system are visualized in Figure 6 (right). Based on those measurements we recommend to use the V-cycle with one pre- and one post-smoothing step, since the resulting condition number remains below *ten* and the resulting preconditioner is symmetric.

4.2. Resolution of parabolic evolution equations. In this section we describe the basic form of the complete algorithm for the space-time resolution of parabolic evolution equations based on the nonsymmetric variational formulation (10), and discuss some of its properties. See [2, 3, 4, 5, 6] for further numerical experiments. We briefly comment on the symmetric one (14) in Section 4.2.6.

4.2.1. Model problem, discretization and setup. As a specific model problem we take the heat equation posed on the L-shaped domain $D = (-1, 1)^2 \setminus [0, 1]^2$ in $d = 2$ spatial dimensions, with the initial datum $g(x) := (1 - |x_1|)(1 - |x_2|)$, right hand side $f(t) := 2t$, and homogeneous Dirichlet boundary conditions. The fact that the initial datum does not satisfy the boundary conditions leads to a solution with low space-time regularity, which is reflected in the convergence rates, see Section 4.2.3.

The spatial domain is partitioned into two pairs of congruent triangles such that $|x_1| - |x_2|$ has constant sign on each triangle. Thereafter, global uniform red refinement is applied resulting in 3, 21, 105, 465, 1'953, 8'001, 32'385 internal degrees of freedom of the P1 Lagrangian finite element discretization.

The temporal interval is partitioned into $N = 2^5, 2^6, \dots, 2^{13}$ temporal elements of equal size. As temporal discretization $E_L \subset H^1(J)$ and $F_L \subset L_2(J)$ we take the

space of continuous piecewise affine and piecewise constant functions with respect to this partition, respectively, which results in $N + 1$ temporal degrees of freedom. This discretization, which is fact equivalent to Crank–Nicholson time-stepping, is not unconditionally space-time stable, more precisely [5]

$$\gamma_L^{-1} \sim 1 + \min\{\sqrt{T}\Lambda_L, \text{CFL}_L\} \quad (55)$$

in (17) for $\Lambda_L := \sup_{\chi \in V_L \setminus \{0\}} \|\chi\|_{H^1} / \|\chi\|_{L_2}$ and $\text{CFL}_L := \Lambda_L^2 T / N$.

The computations are performed on a SGI UV 2000 shared memory machine with 160 cores on 16 CPUs of the Laboratoire J.-L. Lions. The spatial finite element `c++` code used was originally developed in the context of [13]. Linear algebra dense and sparse data structures are provided by the `boost ublas` library with preprocessor macro `NDEBUG` defined, row or column major ordering is used as appropriate. All LU factorizations (disregarding matrix symmetry) are done through `UMFPACK` 4.4 linking to `OpenBLAS` 0.2.14 on default settings and with access to one core per process, whether in parallel or serial computation. Parallelism (described below) is achieved with the `boost mpi` 1.57 library wrapping `OpenMPI` 1.8.3. The code is compiled using `g++` 4.9.2 with the `-O2` optimization flag.

The inverse of $\mathbf{N}$ is computed (only once per MPI process) by LU factorization of the spatial mass/stiffness matrix, see Section 3.2.1. Unless specified otherwise, the application of $\mathbf{M}^{-1}$ is approximated as in Section 3.2.2 using the strategy 2 described following (43) with $\gamma_i^{\text{ref}} = 2^i$ for integer i . Every MPI process keeps its own list of required i , necessitating no more than 11 LU factorizations per process (this number increases as $\log_2 N$ in our setup and decreases with the number of processes). The temporal transformation $\mathbf{V}_t$ is the inverse wavelet transform from Section 3.3 with $\nu = 2$ orthogonalization step, the difference to $\nu = 1$ in the timings is marginal. All LU factorizations are included in the timings shown.

4.2.2. Complete algorithm for the nonsymmetric formulation. The non-symmetric space-time resolution algorithm for parabolic evolution equations proceeds as follows.

1. Assemble the “temporal FEM” matrices $\mathbf{C}_t^{FE}$, $\mathbf{M}_t^{FE}$, $\mathbf{M}_t^E$, $\mathbf{A}_t^E$, $\mathbf{M}_t^F$, and the “spatial FEM” mass and stiffness matrices $\mathbf{M}_x$, $\mathbf{A}_x$, as in Section 3.1. Compute the inverse wavelet transform $\mathbf{T}_t$ as in Section 3.3.1. Assemble the space-time load vector $\mathbf{F}$ as in [4, Section 7.2].
2. Define the functions $\mathbf{w} \mapsto \mathbf{B}\mathbf{w}$ and $\mathbf{d} \mapsto \mathbf{B}^\top \mathbf{d}$ as in Section 3.1.
3. Define the functions that approximate the action $\mathbf{d} \mapsto \mathbf{N}^{-1}\mathbf{d}$ and $\mathbf{w} \mapsto \mathbf{M}^{-1}\mathbf{w}$ of the preconditioners as in Section 3.2.1 and 3.2.2, respectively.
4. Compute an approximate solution to the Gauss normal equations (24) using the generalized LSQR algorithm provided in the Appendix.

4.2.3. Numerical convergence analysis. We document the convergence of the discrete space-time solution after 10 LSQR iterations with respect to the number of temporal and spatial degrees of freedom in the norms of $L_\infty(J; L_2(D))$, $L_2(J; H^1(D))$, and $L_2(J; L_2(D))$ in Figure 7. The discrete solution computed with 20 LSQR iterations at $(2^{13} + 1) \times 32'385 \approx 265\text{M}$ degrees of freedom is taken as the reference solution for error estimation. Since the trial space is defined on a uniform mesh both in space and time, the low regularity of the exact solution implies low convergence rates that can be expected from quasi-optimality (25), see for instance [46, Section 7] for a discussion.

The error of the discrete solution with 8'001 spatial degrees of freedom as a function of the LSQR iteration number is depicted in Figure 8 (left) for different temporal

resolutions. At the first iteration, the influence of the temporal discretization on the discrete inf-sup constant (55) can be seen: increasing the temporal resolution increases the inf-sup constant γ_L , improving the condition number (27) of the preconditioned algebraic system and leading to a smaller error. In Figure 8 (right) we show the same data as a function of walltime computed on 64 MPI processes, see Section 4.2.5.

Figure 9 shows the same computation but with a multigrid iteration replacing the exact LU factorization for the Helmholtz problems (43). The mesh hierarchy is the one generated by the refinement starting from the coarsest mesh with 4 elements. The timings do not include the assembly of the matrices on different levels or their lexicographic reordering. As suggested by the 1d computations in Section 4.1.4, we apply the V-cycle with one pre- and one post-smoothing Gauss–Seidel step. The frequency γ is again rounded to the next power of two (on logarithmic scale). The LU factorization and the multigrid iteration show a similar behavior both in terms of the error and the computational time, except most notably in the initialize phase of the LSQR algorithm where all the necessary LU factorizations are computed.

4.2.4. Data structures and parallelization. The space-time tensor product discretization of the trial space described in Section 3.1 implies that the discrete solution vector $\mathbf{u}$, whose components we index by the basis functions $(\theta, \sigma) \in \Theta \times \Sigma$, can be written as a rectangular array $\mathbf{U} \in \mathbb{R}^{\Sigma \times \Theta}$ where each column corresponds to a spatial vector of size $\#\Sigma$ associated to one of the temporal basis functions $\theta \in \Theta$. We write $\text{Vec}(\mathbf{U}) := \mathbf{u}$. As anticipated in the introduction, for parallel computation we evenly distribute this rectangular array columnwise across the MPI processes. For example, for $2^{10} + 1$ temporal degrees of freedom and 64 MPI processes, the first one hosts 17 columns, all the others 16 columns each. Mutatis mutandis, this applies to the space-time part of the load vector $\mathbf{F}$ and all the vector quantities in the LSQR algorithm. The initial datum part of the load vector $\mathbf{F}$ (and similar quantities in the LSQR algorithm) is assembled and kept on each MPI process. The “spatial FEM” matrices are computed, stored, and LU factorized on demand by each MPI process independently.

The above data layout is tailored to the identity $(\mathbf{S}_t \otimes \mathbf{S}_x) \text{Vec}(\mathbf{U}) = \text{Vec}(\mathbf{S}_x \mathbf{U} \mathbf{S}_t^T)$, where $\mathbf{S}_x$ and $\mathbf{S}_t$ are matrices of appropriate size. Using this identity in the application of $\mathbf{B}$, $\mathbf{N}^{-1}$, etc., allows to avoid the formation of the Kronecker product of matrices. This remains true for the preconditioners $\mathbf{M}$ and $\mathbf{N}$ even if the generator A is time-dependent. Moreover, since the application of $\mathbf{S}_x$ is columnwise, it is performed in parallel. It is thus only in the multiplication by “temporal FEM” matrices and in the wavelet transform that communication between the MPI processes is required. In particular, the application of the approximate inverse of the block-diagonal preconditioner (40) is performed in parallel without any interprocess communication.

4.2.5. Parallel scaling. In this section we comment on the parallel scaling of the complete space-time algorithm from Section 4.2.2. In order to focus on the parallelization in the temporal direction, we fix the number of spatial degrees of freedom to 1’953. We compare the run time of the space-time algorithm on 16, 32, 64, 128 MPI processes with that of a sequential implementation of the implicit Euler time-stepping scheme on the same temporal mesh (the timings for Crank–Nicholson time-stepping are, of course, similar). In each step, the linear system is solved by LU factorization. We point out that the factorization is performed anew for each time step, although the matrix remains the same. This simulates the situation of a diffusion coefficient with a nontrivial temporal dependence or of a nonuniform time step size; otherwise many more MPI processes and time steps, or a finer spatial discretization

would be required to compete with the time-stepping scheme in terms of walltime.

To compare the run times of the space-time algorithm and the time-stepping scheme we do not include mass/stiffness matrix and load vector assembly time, and other elements that are common to both methods. We report on the walltime of the LSQR algorithm with exactly 20 iterations (a rather generous number, see Figures 8 and 9) for the space-time algorithm and essentially the N matrix factorizations that are necessary for time-stepping.

For the wavelet transform in time we use a preliminary parallel implementation of the pyramid scheme, on which we intend to elaborate in a forthcoming work. The application of $\mathbf{T}_t$ and $\mathbf{T}_t^T$ in the approximation of $\mathbf{M}^{-1}$ together require approximately a quarter of the walltime for each LSQR iteration. Our implementation of the wavelet transform through matrix representation and matrix multiplication instead of the pyramid scheme does not scale as well, and we therefore omit the details.

The results are displayed in Figure 10 as a function of the number of MPI processes (left) and as a function of the number of temporal elements (right). The parallel space-time algorithm shows a decent speed up compared to the sequential time-stepping, for instance by a factor of 16 for $N = 2^{12}$ temporal elements on 128 MPI processes. Concerning weak scaling, a computation with 8x the number of MPI processes and 8x the number of temporal intervals takes approximately 2x longer.

4.2.6. On the implementation of the symmetric formulation. The main potential of the symmetric formulation (14) is in that the discretization is uniformly stable in $X_L \subset X$. Moreover, the Lagrangian from which it derives [2, Section 3.2.4] assumes its minimum at the exact solution, and can therefore drive the a posteriori refinement [48]. The different spatial resolutions may be associated with the single scale temporal hat functions (complicating the application of the preconditioner) or directly with the temporal wavelets. In any case, the application of the discretized parabolic operator involves the inverse of (the symmetric part of) the generator A , say the Laplacian, that typically has to be computed on a yet finer spatial mesh up to certain accuracy. This limits the accuracy of the resulting discrete solution but these spatial problems are independent and can therefore be computed in parallel. Details are delegated to future work.

5. Conclusions. We have developed a wavelet-in-time multigrid-in-space preconditioner for algebraic linear systems arising from space-time Petrov–Galerkin discretizations of linear parabolic evolution equations. The sparsity of the wavelet-in-time transformation is crucial to reduce the inter-process communication cost when the parallelization is done along the temporal direction. This transformation approximately block-diagonalizes the canonical preconditioner given by the continuous space-time norms, and allows to invert the resulting spatial blocks in parallel using for instance standard spatial multigrid methods. We have presented several numerical experiments documenting the excellent performance of the preconditioner in the regime of small Péclet numbers, including a first application to robust preconditioning of optimality systems from optimal control constrained by parabolic PDEs, as well as parallel-in-time computations.

Acknowledgment. The author thanks the referees for numerous helpful comments and for pointing out some references. The first draft was written while at RICAM, Austrian Academy of Sciences, Linz (AT). Supported in part by ANR-12-MONU-0013.

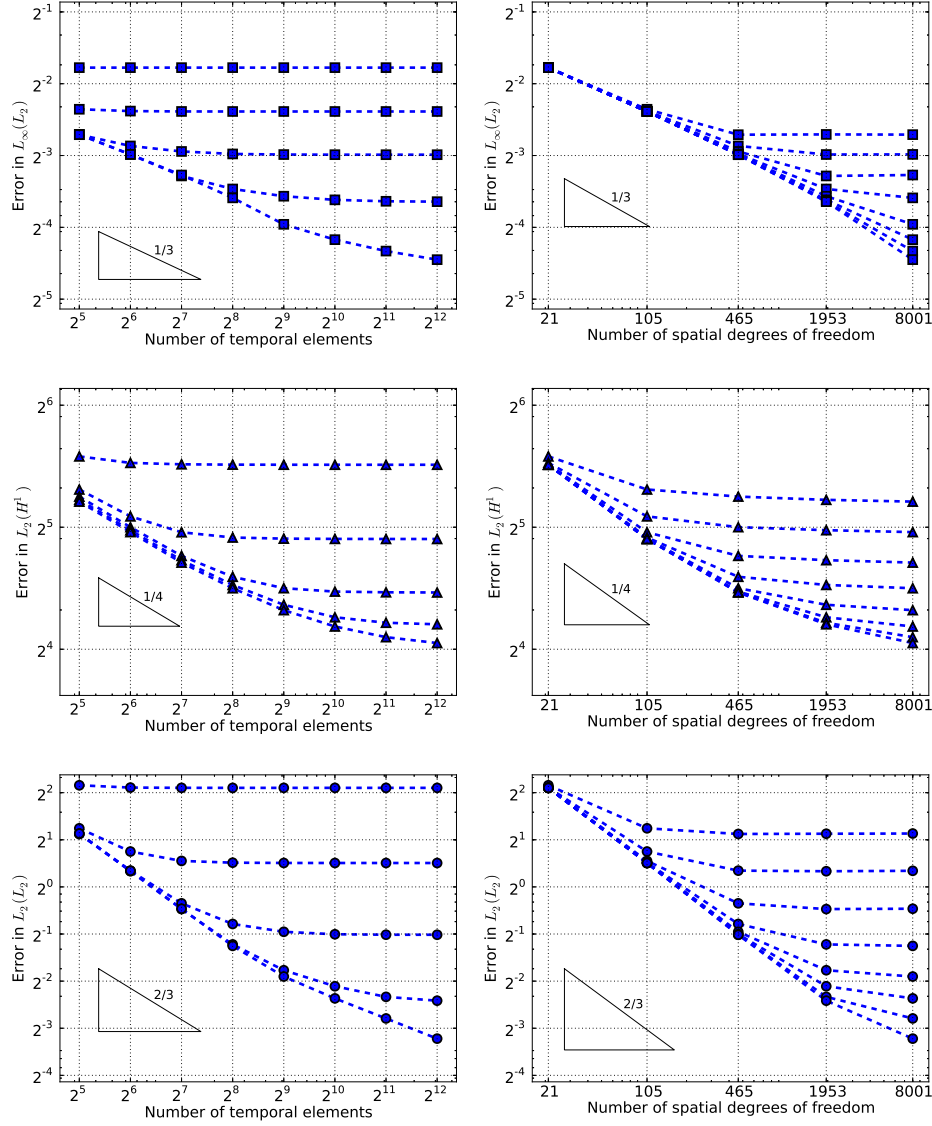


FIG. 7. Error of the approximate discrete space-time solution after 10 LSQR iterations in the setup of Section 4.2.3; top row: in $L_\infty(J; L_2(D))$, middle row: in $L_2(J; H^1(D))$; bottom row: in $L_2(J; L_2(D))$. Left: as a function of the temporal resolution for 21, 105, ..., 8'001 spatial degrees of freedom (top to bottom in each graph). Right: as a function of the spatial resolution for $N = 2^5, \dots, 2^{12}$ temporal elements (top to bottom in each graph).

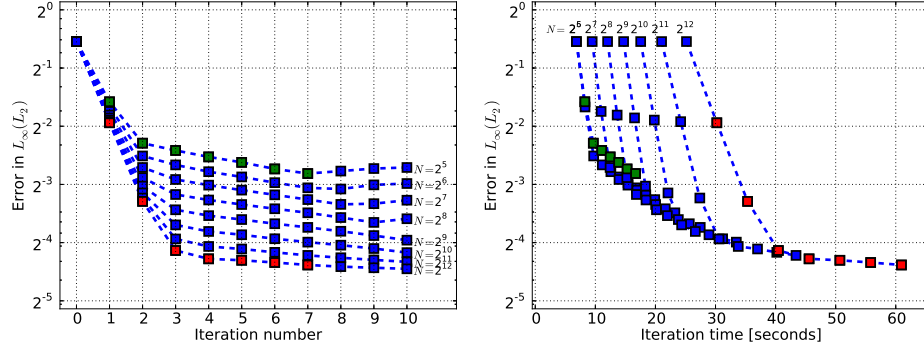


FIG. 8. Error in $L_\infty(J; L_2(D))$ of the approximate discrete solution at step f) of the LSQR iteration for $N = 2^5, \dots, 2^{12}$ temporal elements and 8'001 spatial degrees of freedom in the setup of Section 4.2.3. Left: as a function of the iteration number. Right: first 7 iterations as a function of elapsed walltime from the start of LSQR (computation with 64 MPI processes).

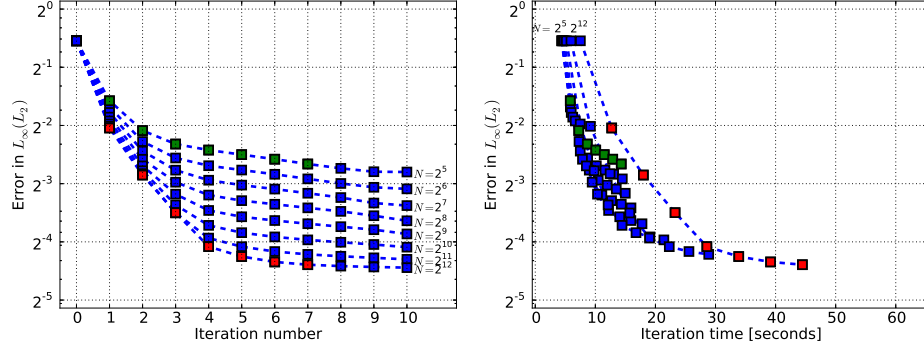


FIG. 9. Same as Figure 8 but with the “V, 1/1” multigrid iteration instead of the direct resolution for the Helmholtz problems.

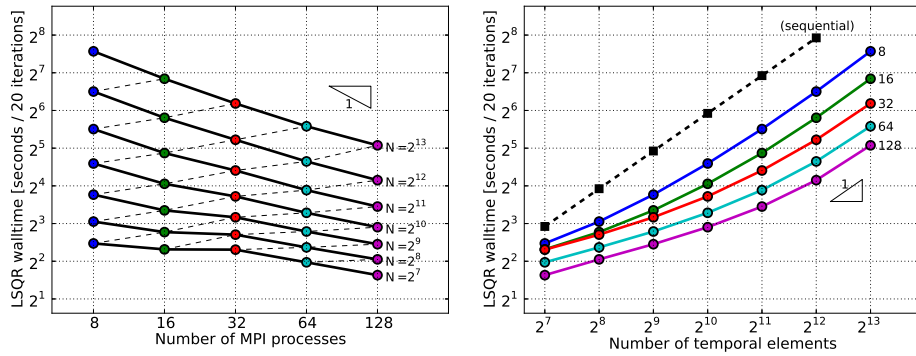


FIG. 10. Walltime of the LSQR algorithm from the start until the completion of 20 iterations for 1'953 spatial degrees of freedom, see Section 4.2.5. Left: as a function of the number of MPI processes for different temporal resolutions; the dashed line indicates weak scaling. Right: as a function of the temporal resolution for different numbers of MPI processes, compared with a sequential implementation of the implicit Euler time-stepping scheme.

Appendix A: Matlab code. Below we include the Matlab code for the construction of the inverse wavelet transform matrix $\mathbf{T}_t$, see Section 3.3.

```

1 % Given: Real vector TE = [ a < b < ... ], real nu > 1
2 % Computes: Inverse wavelet transformation matrix Tt
3
4 % Temporal mass and stiffness matrix on the mesh TE
5 N = length(TE); h = diff(TE); g = 1./h;
6 Mt = spdiags([h 0; 0 h]' * [1 2 0; 0 2 1]/6, -1:1, N, N);
7 At = spdiags([g 0; 0 g]' * [-1 1 0; 0 1 -1], -1:1, N, N);
8
9 % Rows of Tt are coefficients of wavelets wrt finest hats
10 Tt = speye(N);
11
12 % Auxiliary variables corresponding to current coarse level
13 mc = Mt; ac = At; IC = 1:N;
14 while (IC(end) >= 3)
15     % Find most energetic hats
16     eta = 1.9; % Relative energy bandwidth
17     IF = []; e0 = 0;
18     while (length(IC) >= 3)
19         e1 = diag(ac) ./ diag(mc); e1([1 end]) = 0;
20         [e1, j] = max(e1);
21         if (e1 <= e0 / eta); break; end
22         if (e0 == 0); e0 = e1; end
23
24         % Neighbors of the fine hat j get coarsened:
25         J = j + (-1:1);
26         dt = speye(3); dt([1;3], 2) = -ac(j+[-1;1], j) / ac(j, j);
27         ac(J, :) = dt * ac(J, :); ac(:, J) = ac(:, J) * dt';
28         mc(J, :) = dt * mc(J, :); mc(:, J) = mc(:, J) * dt';
29         ac(j, :) = []; ac(:, j) = []; mc(j, :) = []; mc(:, j) = [];
30         Tt(IC(J), :) = dt * Tt(IC(J), :);
31         % There is one more fine hat and one less coarse hat
32         IF = [IF, IC(j)]; IC(j) = [];
33     end
34
35     % Approximate nu-fold orthogonalization
36     P = @(X) X - (X * Mt * Tt(IC,:)) * diag(1./sum(mc)) * Tt(IC,:);
37     for k = 1:nu; Tt(IF,:) = P(Tt(IF,:)); end
38
39     % Reorder rows (first coarse then fine basis functions)
40     Tt(1:IC(end), :) = Tt([IC, IF], :); IC = 1:length(IC);
41 end

```

Appendix B: Generalized LSQR algorithm. We give an adaptation of the LSQR algorithm [44, 11] for the generalized Gauss normal equations $\mathbf{B}^\top \mathbf{N}^{-1} \mathbf{B} \mathbf{u} = \mathbf{B}^\top \mathbf{N}^{-1} \mathbf{F}$ with a preconditioner $\mathbf{M}$. The residual $r_i := \|\mathbf{B}^\top \mathbf{N}^{-1} (\mathbf{B} \mathbf{u}_i - \mathbf{F})\|_{\mathbf{M}^{-1}}$ may be used as a stopping criterion. It is available in each iteration as $r_i = |\delta| \gamma$ following step f). Here, $\text{NORM} : (\mathbf{s}, \mathbf{S}) \mapsto (\hat{\mathbf{z}}, \mathbf{z}, z)$ for a s.p.d. matrix $\mathbf{S}$ is the “normalization” procedure: Approximate $\hat{\mathbf{s}} \approx \mathbf{S}^{-1} \mathbf{s}$, set $z := \sqrt{\mathbf{s}^\top \hat{\mathbf{s}}}$ and $(\hat{\mathbf{z}}, \mathbf{z}) := (z^{-1} \hat{\mathbf{s}}, z^{-1} \mathbf{s})$. The generalized LSQR algorithm consists of an initialization and an iteration phase:

- | | |
|--|--|
| <p>I. Initialize:</p> <p>a) $\mathbf{d} \leftarrow \mathbf{0}$</p> <p>b) $(\hat{\mathbf{v}}, \mathbf{v}, \beta) \leftarrow \text{NORM}(\mathbf{F}, \mathbf{N})$</p> <p>c) $(\hat{\mathbf{w}}, \mathbf{w}, \alpha) \leftarrow \text{NORM}(\mathbf{B}^\top \hat{\mathbf{v}}, \mathbf{M})$</p> <p>d) $\rho \leftarrow \ (\alpha, \beta)\ _2$</p> <p>e) $\mathbf{u}_0 \leftarrow \mathbf{0}$</p> <p>f) $(\delta, \gamma) \leftarrow (\alpha, \beta)$</p> | <p>II. Iterate for $i = 1, 2, \dots$ (until convergence):</p> <p>a) $\mathbf{d} \leftarrow \hat{\mathbf{w}} - (\alpha\beta/\rho^2)\mathbf{d}$</p> <p>b) $(\hat{\mathbf{v}}, \mathbf{v}, \beta) \leftarrow \text{NORM}(\mathbf{B}\hat{\mathbf{w}} - \alpha\mathbf{v}, \mathbf{N})$</p> <p>c) $(\hat{\mathbf{w}}, \mathbf{w}, \alpha) \leftarrow \text{NORM}(\mathbf{B}^\top \hat{\mathbf{v}} - \beta\mathbf{w}, \mathbf{M})$</p> <p>d) $\rho \leftarrow \ (\delta, \beta)\ _2$</p> <p>e) $\mathbf{u}_i \leftarrow \mathbf{u}_{i-1} + (\delta\gamma/\rho^2)\mathbf{d}$</p> <p>f) $(\delta, \gamma) \leftarrow (-\delta\alpha/\rho, \gamma\beta/\rho)$</p> |
|--|--|

REFERENCES

- [1] J. M. Alam, N. K.-R. Kevlahan, and O. V. Vasilyev. Simultaneous space-time adaptive wavelet solution of nonlinear parabolic differential equations. *J. Comput. Phys.*, 214(2):829–857, 2006.
- [2] R. Andreev. *Stability of space-time Petrov-Galerkin discretizations for parabolic evolution equations*. PhD thesis, ETH Zürich, 2012. ETH Diss. No. 20842.
- [3] R. Andreev. Stability of sparse space-time finite element discretizations of linear parabolic evolution equations. *IMA J. Numer. Anal.*, 33(1):242–260, 2013.
- [4] R. Andreev. Space-time discretization of the heat equation. *Numer. Algorithms*, 67(4):713–731, 2014.
- [5] R. Andreev and J. Schweitzer. Conditional space-time stability of collocation Runge–Kutta for parabolic evolution equations. *Electron. Trans. Numer. Anal.*, 41:62–80, 2014.
- [6] R. Andreev and C. Tobler. Multilevel preconditioning and low rank tensor iteration for space-time simultaneous discretizations of parabolic PDEs. *Numer. Lin. Algebra. Appl.*, 22(2):317–337, 2015.
- [7] I. Babuška and T. Janik. The h-p version of the finite element method for parabolic equations. I. The p version in time. *Numer. Meth. Part. D. E.*, 5:363–399, 1989.
- [8] I. Babuška and T. Janik. The h-p version of the finite element method for parabolic equations. II. The h-p version in time. *Numer. Meth. Part. D. E.*, 6:343–369, 1990.
- [9] L. Banjai and D. Peterseim. Parallel multistep methods for linear evolution problems. *IMA J. Numer. Anal.*, 32(3):1217–1240, 2012.
- [10] R. E. Bank and H. Yserentant. On the H^1 -stability of the L_2 -projection onto finite element spaces. *Numer. Math.*, 126(2):361–381, 2014.
- [11] S. J. Benbow. Solving generalized least-squares problems with LSQR. *SIAM J. Matrix Anal. Appl.*, 21(1):166–177, 1999.
- [12] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numer.*, 14:1–137, 2005.
- [13] M. Bieri, R. Andreev, and C. Schwab. Sparse tensor discretization of elliptic spdes. *SIAM J. on Sci. Comput.*, 31(6):4281–4304, 2009/10.
- [14] J. H. Bramble, J. E. Pasciak, and P. S. Vassilevski. Computational scales of Sobolev norms with application to preconditioning. *Math. Comp.*, 69(230):463–480, 2000.
- [15] H. Brézis and I. Ekeland. Un principe variationnel associé à certaines équations paraboliques. Le cas dépendant du temps. *C. R. Acad. Sci. Paris Sér. A-B*, 282(20):A1, A1197–A1198, 1976.
- [16] N. Chegini and R. Stevenson. Adaptive wavelet schemes for parabolic problems: Sparse matrices and numerical results. *SIAM J. Numer. Anal.*, 49(1):182–212, 2011.
- [17] N. Chegini and R. Stevenson. The adaptive tensor product wavelet scheme: sparse matrices and the application to singularly perturbed problems. *IMA Journal of Numerical Analysis*, 32(1):75–104, 2012.
- [18] A. Chernov and C. Schwab. Sparse space-time Galerkin BEM for the nonstationary heat equation. *Z. Angew. Math. Mech.*, 93(6-7):403–413, 2013.
- [19] A. Cohen, I. Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 45(5):485–560, 1992.
- [20] M. Costabel. Boundary integral operators for the heat equation. *Integral Equations Operator Theory*, 13(4):498–552, 1990.
- [21] S. V. Dolgov, B. N. Khoromskij, and I. V. Oseledets. Fast solution of parabolic problems in the tensor train/quantized tensor train format with initial application to the Fokker-Planck equation. *SIAM J. Sci. Comput.*, 34(6):A3016–A3038, 2012.
- [22] L. C. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, 1998.
- [23] M. J. Gander. 50 years of time parallel time integration. In T. Carraro, M. Geiger, S. Körkel, and R. Rannacher, editors, *Multiple shooting and time domain decomposition*. Springer, to appear.
- [24] M. J. Gander and M. Neumüller. Analysis of a new space-time parallel multigrid algorithm for parabolic problems. ArXiv 1411.0519, 2014.
- [25] M. J. Gander and S. Vandewalle. Analysis of the parareal time-parallel time-integration method. *SIAM J. Sci. Comput.*, 29(2):556–578, 2007.
- [26] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.
- [27] M. Griebel and D. Oeltz. A sparse grid space-time discretization scheme for parabolic problems. *Computing*, 81(1):1–34, 2007.

- [28] M. Griebel, D. Oeltz, and P. Vassilevski. Space-time approximation with sparse grids. *SIAM J. Sci. Comput.*, 28(2):701–727, 2006.
- [29] W. Hackbusch. Fast numerical solution of time-periodic parabolic problems by a multigrid method. *SIAM J. Sci. Statist. Comput.*, 2(2):198–206, 1981.
- [30] W. Hackbusch. *Multigrid methods and applications*, volume 4. Springer-Verlag, Berlin, 1985. Second printing 2003.
- [31] L. Halpern. Schwarz waveform relaxation algorithms. In *Domain decomposition methods in science and engineering XVII*, volume 60 of *Lect. Notes Comput. Sci. Eng.*, pages 57–68. Springer, Berlin, 2008.
- [32] G. Horton and S. Vandewalle. A space-time multigrid method for parabolic partial differential equations. *SIAM J. Sci. Comput.*, 16(4):848–864, 1995.
- [33] J. Janssen and S. Vandewalle. Multigrid waveform relaxation of spatial finite element meshes: the continuous-time case. *SIAM J. Numer. Anal.*, 33(2):456–474, 1996.
- [34] S. Kestler, K. Steih, and K. Urban. An efficient space-time adaptive wavelet Galerkin method for time-periodic parabolic partial differential equations. arXiv 1401.5782, 2014.
- [35] M. Kiehl. Parallel multiple shooting for the solution of initial value problems. *Parallel Comput.*, 20(3):275–295, 1994.
- [36] U. Langer and M. Wolfram. Multiharmonic finite element analysis of a time-periodic parabolic optimal control problem. *J. Numer. Math.*, 21(4):265–300, 2013.
- [37] S. Larsson and C. Schwab. Compressive space-time Galerkin discretizations of parabolic partial differential equations. SAM Report 2015-04, ETH Zürich, 2015.
- [38] J.-L. Lions, Y. Maday, and G. Turinici. Résolution d’EDP par un schéma en temps “pararéel”. *C. R. Acad. Sci. Paris Sér. I Math.*, 332(7):661–668, 2001.
- [39] J.-L. Lions and E. Magenes. *Non-homogeneous boundary value problems and applications. Vol. I*. Springer-Verlag, New York, 1972.
- [40] C. Mollet. Stability of Petrov–Galerkin discretizations: application to the space-time weak formulation for parabolic evolution problems. *Comput. Methods Appl. Math.*, 14(2):135–303, 2014.
- [41] B. Nayroles. Deux théorèmes de minimum pour certains systèmes dissipatifs. *C. R. Acad. Sci. Paris Sér. A-B*, 282(17):Aiv, A1035–A1038, 1976.
- [42] M. Neumüller and O. Steinbach. A DG space-time domain decomposition method. In R. Bank, M. Holst, O. Widlund, and J. Xu, editors, *Domain Decomposition Methods in Science and Engineering XX*, volume 91 of *LNCSE*, pages 623–630. Springer, 2013.
- [43] M. A. Olshanskii and A. Reusken. On the convergence of a multigrid method for linear reaction-diffusion problems. *Computing*, 65(3):193–202, 2000.
- [44] C. C. Paige and M. A. Saunders. LSQR: an algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Software*, 8(1):43–71, 1982.
- [45] J. W. Pearson, M. Stoll, and A. J. Wathen. Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems. *SIAM J. Matrix Anal. Appl.*, 33(4):1126–1152, 2012.
- [46] C. Schwab and R. Stevenson. Space-time adaptive wavelet methods for parabolic evolution problems. *Math. Comp.*, 78(267):1293–1318, 2009.
- [47] D. Sheen, I. H. Sloan, and V. Thomée. A parallel method for time discretization of parabolic equations based on Laplace transformation and quadrature. *IMA J. Numer. Anal.*, 23(2):269–299, 2003.
- [48] U. Stefanelli. The discrete Brezis-Ekeland principle. *J. Convex Anal.*, 16(1):71–87, 2009.
- [49] F. Tantardini. *Quasi-optimality in the backward Euler-Galerkin method for linear parabolic problems*. PhD thesis, Università degli Studi di Milano, 2014.
- [50] S. Vandewalle and G. Horton. Fourier mode analysis of the multigrid waveform relaxation and time-parallel multigrid methods. *Computing*, 54(4):317–330, 1995.
- [51] P. S. Vassilevski and J. Wang. Stabilizing the hierarchical basis by approximate wavelets. I. Theory. *Numer. Lin. Algebra. Appl.*, 4(2):103–126, 1997.
- [52] P. S. Vassilevski and J. Wang. Stabilizing the hierarchical basis by approximate wavelets. II. Implementation and numerical results. *SIAM J. Sci. Comput.*, 20(2):490–514, 1998.
- [53] A. J. Wathen. Preconditioning. *Acta Numer.*, 24:329–376, 2015.
- [54] W. Zulehner. Nonstandard norms and robust estimates for saddle point problems. *SIAM J. Matrix Anal. Appl.*, 32(2):536–560, 2011.