Electronic Transactions on Numerical Analysis.
Volume 45, pp. 283–304, 2016.

ISSN 1068–9613.

ETNA
Kent State University
http://etna.math.kent.edu

A MATRIX-FREE LEGENDRE SPECTRAL METHOD FOR INITIAL-BOUNDARY VALUE PROBLEMS*

BERND BRUMM† AND EMIL KIERI‡

Abstract. We present a Legendre spectral method for initial-boundary value problems with variable coefficients and of arbitrary dimensionality, where the computational work in each time step scales linearly with the number of unknowns. Boundary conditions are enforced weakly, allowing for stable solutions of many classes of problems. Working in coefficient space, derivatives can be evaluated recursively in linear time. We show how also the action of variable coefficients can be implemented without transforming back to coordinate space using a recursive, linearly scaling matrix-free algorithm, under the assumption that the coefficients vary on a much longer scale than the solution. We also prove that spectral accuracy is preserved for smooth solutions. Numerical results for the wave equation in two and three dimensions corroborate the theoretical predictions.

Key words. spectral methods, matrix-free methods, method of lines, stability, computational wave propagation, boundary conditions

AMS subject classifications. 65M12, 65M15, 65M20, 65M70

1. Introduction.

For smooth solutions, spectral and pseudospectral spatial discretisations of partial differential equations (PDEs) converge faster than any polynomial in the number of basis functions [23, 33]. This is to be contrasted to the algebraic convergence rate of standard finite difference and finite element methods (FEM). The price for this formally infinite order of accuracy is the globality of the discretisation. The basis functions extend over the entire domain, and the resulting differentiation matrices are in general dense. Done naively, the cost of computing derivatives by matrix-vector multiplication therefore scales quadratically with the number of unknowns, and the assembly of the dense matrices is even more expensive. The higher computational complexity of spectral methods is a considerable disadvantage as compared to the linearly scaling finite difference methods and FEM, which correspond to differentiation matrices of finite bandwidth. In this paper, we present a means of circumventing this disadvantage. For Fourier and Chebyshev methods, the fast Fourier transform (FFT) provides a loophole; see [29, Ch. 12]. In coefficient space, the expansions can be differentiated in linear time. Multiplication by variable coefficients scales linearly in physical space, and the FFT is an essentially linearly scaling means for going between the two spaces. There are, unfortunately, no fast transforms for most spectral bases. This is problematic in particular for problems with non-trivial boundary conditions, where Fourier and Chebyshev methods are inappropriate. In the present work, we propose a spectral discretisation for initial-boundary value problems in a Legendre basis, using weak enforcement of boundary conditions. To specify its main features, our matrix-free methodology employs no transformations, allows for variable coefficients, and avoids limitations on what kinds of boundary conditions are permitted. Still, evaluating the action of both derivative and variable coefficient matrices onto vectors scales essentially linearly.

There are two main approaches to enforce boundary conditions: Strongly, where the solution is set to satisfy the boundary conditions exactly, and weakly, where penalty terms force the solution to satisfy the boundary conditions approximately. Strong enforcement of boundary

*Received December 18, 2015. Accepted June 25, 2016. Published online on July 29, 2016. Recommended by J. A. C. Weideman.

†Mathematisches Institut, Universität Tübingen, Auf der Morgenstelle 10, D-72076 Tübingen, Germany (brumm@na.uni-tuebingen.de).

‡Division of Scientific Computing, Department of Information Technology, Uppsala University, Box 337, 751 05 Uppsala, Sweden. Present address: Hausdorff Center for Mathematics & Institute for Numerical Simulation, University of Bonn, 53115 Bonn, Germany (kieri@ins.uni-bonn.de).

ETNA
Kent State University
http://etna.math.kent.edu

conditions may seem more natural, and for problems with certain simple boundary conditions and constant coefficients problem-specific bases which satisfy the boundary conditions can be constructed [32]. More general boundary conditions can be implemented strongly using boundary bordering [3, Ch. 6]. This however breaks the structure of the matrix, and stability of the scheme can be an issue. Weak enforcement of boundary conditions using penalties was introduced for spectral methods in [14, 15]. Penalty methods are flexible, and can be applied to a good variety of boundary conditions for PDEs of different type. By construction, the resulting spatial discretisations satisfy similar energy estimates as the continuous problems, which in combination with suitable time stepping guarantees stability. Penalty methods are also essential building blocks for discontinuous Galerkin (DG) methods [10, 20, 26], and the dominant way of imposing boundary conditions for finite difference methods on summation-by-parts (SBP) form [7, 24, 30]. Weak boundary conditions are typically implemented analogously for DG, SBP and spectral methods. In the present work, we employ a spectral penalty approach.

For spectral penalty methods, one typically uses a Lagrange basis on Gauß–Lobatto quadrature nodes [6, 13, 18, 19]. The motivation is that if the corresponding quadrature rule is used to evaluate the integrals defining it, the mass matrix becomes diagonal. This simplifies the implementation and improves the efficiency of the method, not least by facilitating explicit time stepping. Differentiation matrices are however still dense, and the method therefore scales quadratically with the number of unknowns. A way to speed up the computations is sum-factorisation [25, 28], which takes advantage of the tensor product structure of the grid. This reduces the computational work from $\mathcal{O}(K^{2d})$ to $\mathcal{O}(K^{d+1})$, where K is the number of degrees of freedom per dimension. Another technique, developed in the context of spectral approximations to the Schrödinger equation, is the fast algorithm due to [4, 12], which allows to compute the action of the Hamiltonian on a vector in linear time only, i.e., $\mathcal{O}(K^d)$. In the present work, we modify and adapt the latter technique for initial-boundary value problems.

A brief sketch of our main ideas goes along the following lines: As far as we know, one cannot use weak boundary conditions and at the same time take advantage of diagonal mass matrices and the fast Fourier transform. The present approach combines the flexibility of weak enforcement of boundary conditions with the convenience of a standard Legendre basis and still yields linear costs in every time step. We advocate working in the Legendre coefficient space instead of in a nodal basis, and, in higher dimensions, tensor products of univariate Legendre polynomials are employed. This choice allows to compute derivatives in linear time using well-known recurrence relations of the Legendre polynomials. In Legendre space, however, variable coefficients lead to dense matrices, which have to be multiplied to the vector of Legendre coefficients in every time step. Still, as we show, the matrix-vector products can also be evaluated efficiently in essentially linear time, provided that the coefficients are much smoother than the solution. This is a typical situation: For wave propagation problems, or in the presence of boundary layers, the solution can exhibit rapid oscillations despite the coefficients of the PDE being smooth. These are problems where spectral and high order methods are advantageous. In the present work, the action of coefficient matrices on a vector is computed through an approach based on orthogonality and recurrence relations of the Legendre polynomials, using polynomial approximation of the variable coefficient functions. The idea of using the properties of orthogonal polynomials to solve differential equations with polynomial coefficients was first introduced in [8]. It was elaborated on in more detail in [9], and more recently in [27, 35]. These works primarily consider boundary value problems. By approximating variable coefficients by polynomials they get banded discretisation matrices, and thereby linearly scaling methods. They enforce boundary conditions, or more general linear constraints, using boundary bordering.

ETNA
Kent State University
http://etna.math.kent.edu

The technique we consider in this paper is similar, but more explicitly designed for time-dependent problems. It was first mentioned in [12], and analysed in more detail in [4], in a basis of Hermite functions. We modify this technique and, in particular, adapt it to a Legendre basis and problems with boundaries. On the whole, we compute all occurring matrix-vector products in linear time without actually assembling any matrices or performing explicit matrix-vector multiplications. This constitutes a considerable gain in efficiency. On the disadvantage side, we still have a quadratic cost for transformations between physical and coefficient space. We need such transforms, e.g., to prepare the initial data and to visualise the solution. These operations are, however, rare compared to the evaluation of the differential operator, which needs to be done in every time step.

Related work has also been done in the context of p- and hp-FEM [1, 31]. In [2], a basis of integrated Jacobi polynomials was used to construct an hp-FEM with linear complexity for elliptic problems with (piecewise) constant coefficients. In [22], they used recurrence relations to evaluate derivatives in linear time, and sum-factorisation to speed up the treatment of variable material properties.

The remainder of this paper is organised as follows. In Section 2, we present our Legendre–Galerkin spectral method. For illustration purposes, we use the example of a wave equation with spatially variable coefficients on the unit hypercube. In Section 3, we explain how to compute the different kinds of matrix-vector products in linear time, without assembling any matrices. As it turns out, these efficient procedures are related to Gauß–Legendre quadrature. In Section 4, we briefly derive energy estimates both for the exact solution and for the spectral method by means of standard techniques, proving stability of the scheme. Using new ideas, an a priori error estimate for the spectral discretisation is given in Section 5. In particular, we analyse the error caused by polynomial interpolation of the coefficient functions, truncation of the basis, and by Gaussian quadrature. Numerical experiments for the wave equation in curvilinear coordinates are presented in Section 6.

2. The Legendre–Galerkin spectral method. We introduce our method using the educational example of the scalar wave equation with variable coefficients on a d-dimensional hypercube. More general geometries can be handled using curvilinear coordinate transformations. The methodology is general and can be adapted to other well-posed initial-boundary value problems as well. The problem we consider reads

$$u_{tt} = \nabla \cdot a \nabla u + f, \qquad x \in \Omega,\ t \geq 0, \tag{2.1}$$

$$u(x,0) = u_1(x), \qquad x \in \Omega, \tag{2.2}$$

$$u_t(x,0) = u_2(x), \qquad x \in \Omega, \tag{2.3}$$

$$b u_t + \mathbf{n} \cdot a \nabla u = g(x,t), \qquad x \in \partial\Omega,\ t \geq 0, \tag{2.4}$$

with $a = a^T$ positive definite, $\det(a(x)) \geq a_0 > 0$, and $b(x) \geq b_0 > 0$ for all $x \in \Omega = [-1,1]^d$. $\mathbf{n}$ denotes the outward unit normal. We also assume that the norm and elements of a are bounded for all x, i.e., $\|a(x)\| \leq a_{\max}$, $|a^{(\alpha,\beta)}(x)| \leq a_{\max}$, and that $b(x) \leq b_{\max}$. Throughout, we denote the standard L^2 inner product and norm by $(\cdot,\cdot)$ and $\|\cdot\|$, respectively. Similarly, $(\cdot,\cdot)_{\partial\Omega}$ and $\|\cdot\|_{\partial\Omega}$ denote the L^2 inner product and norm on the boundary.

2.1. Galerkin approximation. We use a Galerkin approach over L^2-normalised Legendre polynomials. In one dimension, they are defined recursively as

$$\begin{aligned} \varphi_k(x) &= \frac{\sqrt{(2k-1)(2k+1)}}{k} x \varphi_{k-1}(x) - \frac{k-1}{k}\sqrt{\frac{2k+1}{2k-3}}\varphi_{k-2}(x), \quad k \geq 2, \\ \varphi_0(x) &= \frac{1}{\sqrt{2}}, \qquad \varphi_1(x) = \sqrt{\frac{3}{2}}x \end{aligned} \tag{2.5}$$

ETNA
Kent State University
http://etna.math.kent.edu

on $[-1,1]$. This normalisation gives an identity mass matrix, which simplifies both the presentation and implementation. The classical Legendre polynomials, $P_k(x)$, are normalised with respect to their boundary values, and relate to our basis functions as

$$\varphi_k(x) = \sqrt{(2k+1)/2}P_k(x).$$

In higher dimensions, we expand the solution in the basis functions $\varphi_{\mathbf{k}}$, $\mathbf{k} \in \mathcal{K} \subset \mathbb{N}^d$, constructed as tensor products of the univariate Legendre polynomials. We denote the Galerkin approximation by

$$v(x,t) = \sum_{\mathbf{k}\in\mathcal{K}} v_{\mathbf{k}}(t)\varphi_{\mathbf{k}}(x), \qquad \varphi_{\mathbf{k}}(x) = \prod_{\alpha=1}^{d} \varphi_{k_\alpha}(x_\alpha),$$

where the multi-indices $\mathbf{k}$ belong to the index set

$$\mathcal{K}(d,K) = \left\{\mathbf{k} \in \mathbb{N}^d \,:\, 0 \le k_\alpha \le K\right\},$$

which is of size $|\mathcal{K}| = (K+1)^d$. For the sake of brevity, we shall often omit the arguments d and K. We denote the polynomial approximation space by $\mathcal{V}_{\mathcal{K}} = \operatorname{span}(\varphi_{\mathbf{k}})_{\mathbf{k}\in\mathcal{K}}$. If we multiply (2.1), with v in the place of u, by a basis function $\varphi_{\mathbf{j}} \in \mathcal{V}_{\mathcal{K}}$ and integrate, we get

$$\sum_{\mathbf{k}\in\mathcal{K}} \ddot{v}_{\mathbf{k}}\,(\varphi_{\mathbf{j}},\varphi_{\mathbf{k}}) = \sum_{\mathbf{k}\in\mathcal{K}} v_{\mathbf{k}}\,(-(\nabla\varphi_{\mathbf{j}}, a\nabla\varphi_{\mathbf{k}}) + (\varphi_{\mathbf{j}}, \mathbf{n}\cdot a\nabla\varphi_{\mathbf{k}})_{\partial\Omega}) + (\varphi_{\mathbf{j}}, f) \tag{2.6}$$

for all $\mathbf{j} \in \mathcal{K}$. Since the basis is orthonormal, the mass matrix $(\varphi_{\mathbf{j}},\varphi_{\mathbf{k}})_{\mathbf{j},\mathbf{k}\in\mathcal{K}}$ is the identity matrix. Eq. (2.6) defines a system of ordinary differential equations (ODEs) for $\mathbf{v} = (v_{\mathbf{k}})_{\mathbf{k}\in\mathcal{K}}$, but it does not respect the boundary conditions and it is not stable. We address this by replacing $\mathbf{n}\cdot\nabla v$ using the boundary condition (2.4), as is commonly done in Galerkin methods. The resulting system of ODEs reads

$$(\phi,\ddot{v}) = -(\nabla\phi, a\nabla v) - (\phi, b\dot{v})_{\partial\Omega} + (\phi, f) + (\phi, g)_{\partial\Omega} \quad \forall\phi \in \mathcal{V}_{\mathcal{K}},$$

or, equivalently,

$$\ddot{\mathbf{v}} = -\mathbf{S}\mathbf{v} - \mathbf{B}\dot{\mathbf{v}} + \mathbf{f} + \mathbf{g}, \tag{2.7}$$

with

$$\begin{aligned} \mathbf{S}_{\mathbf{j},\mathbf{k}} &= \sum_{\alpha,\beta=1}^{d} \mathbf{S}^{(\alpha,\beta)}_{\mathbf{j},\mathbf{k}} = \sum_{\alpha,\beta=1}^{d} \left(\frac{\partial}{\partial x_\alpha}\varphi_{\mathbf{j}}, a^{(\alpha,\beta)}\frac{\partial}{\partial x_\beta}\varphi_{\mathbf{k}}\right), \qquad \mathbf{j},\,\mathbf{k} \in \mathcal{K}. \\ \mathbf{B}_{\mathbf{j},\mathbf{k}} &= (\varphi_{\mathbf{j}}, b\varphi_{\mathbf{k}})_{\partial\Omega}, \qquad \mathbf{f}_{\mathbf{j}} = (\varphi_{\mathbf{j}}, f), \qquad \mathbf{g}_{\mathbf{j}} = (\varphi_{\mathbf{j}}, g)_{\partial\Omega}, \end{aligned} \tag{2.8}$$

We will prove stability for this system in Section 4.

The replacement of the boundary term above can also be seen as adding penalty terms to the scheme in order to enforce the boundary conditions weakly. We did indeed add

$$(\varphi_{\mathbf{j}}, g)_{\partial\Omega} - (\varphi_{\mathbf{j}}, \mathbf{n}\cdot a\nabla v)_{\partial\Omega} + (\varphi_{\mathbf{j}}, b\dot{v})_{\partial\Omega}$$

to the right-hand side of (2.6). These terms vanish when the boundary conditions are satisfied and penalise deviation from the boundary conditions. In general, for linear problems, if the added penalty terms (i) vanish when $v(x,t)$ satisfies the boundary conditions, and (ii) make the semidiscretisation stable, we can expect the scheme to converge.

ETNA
Kent State University
http://etna.math.kent.edu

When the system of ODEs (2.7) is solved using an explicit Runge–Kutta or multistep method, we need to compute the matrix-vector products $\mathbf{S}\mathbf{v}$ and $\mathbf{B}\dot{\mathbf{v}}$ in each time step. Since $\mathbf{S}$ and $\mathbf{B}$ are full matrices, the cost of computing these matrix-vector products is quadratic in $|\mathcal{K}|$. Assembling $\mathbf{S}$ and $\mathbf{B}$ is even more expensive. The novelty of our method is a means to compute spectrally accurate approximations of these products in linear time only, without assembling the matrices explicitly. In Sections 2.3 and 2.4, we approximate and rewrite the products $\mathbf{S}\mathbf{v}$ and $\mathbf{B}\dot{\mathbf{v}}$ appropriately. The linearly scaling procedures to compute the resulting expressions efficiently are presented in Section 3. Section 2.5 states the spatially discretised system which we actually propagate in time.

2.2. Remark on other boundary conditions. In the presentation and analysis of the present method we use the fairly general class of boundary conditions (2.4). Another important class of boundary conditions is Dirichlet boundary conditions. We here briefly mention how they can be enforced in our approach. We then consider the wave equation (2.1)–(2.3) with

$$u = g(x,t), \qquad x \in \partial\Omega,\ t \geq 0.$$

We again seek an approximate solution $v(\cdot,t) \in \mathcal{V}_\mathcal{K}$, and enforce the boundary conditions weakly. Weak enforcement of Dirichlet boundary conditions for the scalar wave equation is discussed in, e.g., [16, 24]. The approximation v satisfies the system of ODEs

$$\begin{aligned}(\phi,\ddot{v}) = &-(\nabla\phi, a\nabla v) + (\phi, \mathbf{n}\cdot a\nabla v)_{\partial\Omega} + (\phi, f)\\ &+ (\mathbf{n}\cdot a\nabla\phi, v-g)_{\partial\Omega} - \sigma(\phi, v-g)_{\partial\Omega} \quad \forall\phi\in\mathcal{V}_\mathcal{K},\end{aligned} \tag{2.9}$$

where $\sigma > 0$ is a sufficiently large scalar which depends on a, Ω and K. The second line in (2.9) contains the penalty terms which weakly enforce the Dirichlet boundary conditions. The system can equivalently be written as

$$\ddot{\mathbf{v}} = -\mathbf{S}\mathbf{v} + \mathbf{B}^{(D)}\mathbf{v} + \mathbf{f} + \mathbf{g}^{(D)},$$

where $\mathbf{S}$ and $\mathbf{f}$ are defined as in (2.8) and the boundary terms are

$$\begin{aligned}\mathbf{B}^{(D)}_{\mathbf{j},\mathbf{k}} &= (\varphi_\mathbf{j}, \mathbf{n}\cdot a\nabla\varphi_\mathbf{k})_{\partial\Omega} + (\mathbf{n}\cdot a\nabla\varphi_\mathbf{j}, \varphi_\mathbf{k})_{\partial\Omega} - \sigma(\varphi_\mathbf{j},\varphi_\mathbf{k})_{\partial\Omega},\\ \mathbf{g}^{(D)}_\mathbf{j} &= -(\mathbf{n}\cdot a\nabla\varphi_\mathbf{j}, g)_{\partial\Omega} + \sigma(\varphi_\mathbf{j}, g)_{\partial\Omega}.\end{aligned}$$

The techniques devised in this paper for solving (2.1)–(2.4) efficiently can be adapted in a straightforward way to handle also this problem. The convergence analysis of problems with Dirichlet boundary conditions is however beyond the scope of this paper.

Neumann boundary conditions are also not covered by the analysis due to the positivity condition we impose on b in (2.4). We do however use them in our numerical experiments without problems. Standard Robin boundary conditions can also be implemented in a straightforward way.

2.3. Approximation of matrix-vector products. We start with some general considerations about Gaussian quadrature, defining

$$\omega_\mathbf{j} = \omega_{j_1}\cdot\ldots\cdot\omega_{j_d}, \qquad \xi_\mathbf{j} = (\xi_{j_1},\ldots,\xi_{j_d}), \qquad \mathbf{j}\in\mathcal{K},$$

where ξ_j and ω_j are the nodes and weights of the $(K+1)$-node Gauß–Legendre quadrature rule. This yields a d-dimensional quadrature rule which is exact for polynomials of degree up to $2K+1$ in each variable. We define the matrices

$$\mathbf{U}_{\mathbf{j},\mathbf{k}} = \sqrt{\omega_\mathbf{j}}\varphi_\mathbf{k}(\xi_\mathbf{j}), \qquad \mathbf{U}^{(\alpha)}_{\mathbf{j},\mathbf{k}} = \sqrt{\omega_\mathbf{j}}\frac{\partial}{\partial x_\alpha}\varphi_\mathbf{k}(\xi_\mathbf{j}), \qquad \mathbf{j},\mathbf{k}\in\mathcal{K},\ \alpha = 1,\ldots,d, \tag{2.10}$$

ETNA
Kent State University
http://etna.math.kent.edu

which satisfy

$$\mathbf{U}^T\mathbf{U} = I, \qquad \mathbf{U}^T\mathbf{U}^{(\alpha)} = \mathbf{D}^{(\alpha)}, \tag{2.11}$$

where $\mathbf{D}^{(\alpha)}$ is the Galerkin differentiation matrix,

$$\mathbf{D}^{(\alpha)}_{\mathbf{j},\mathbf{k}} = \left(\varphi_{\mathbf{j}}, \frac{\partial}{\partial x_\alpha}\varphi_{\mathbf{k}}\right), \qquad \mathbf{j},\mathbf{k} \in \mathcal{K}. \tag{2.12}$$

Given an arbitrary multivariate polynomial $q\colon \Omega \to \mathbb{R}$, we consider the matrices

$$\mathbf{Q}_{\mathbf{j},\mathbf{k}} = (\varphi_{\mathbf{j}}, q\varphi_{\mathbf{k}}), \quad \mathbf{Q}^{(\alpha,\beta)}_{\mathbf{j},\mathbf{k}} = \left(\frac{\partial}{\partial x_\alpha}\varphi_{\mathbf{j}}, q\frac{\partial}{\partial x_\beta}\varphi_{\mathbf{k}}\right), \qquad \mathbf{j},\mathbf{k} \in \mathcal{K}, \quad \alpha,\beta = 1,\ldots,d.$$

Due to (2.11), the matrices $\mathbf{U}$ and $\mathbf{U}^{(\alpha)}$ can be used to denote the quadrature approximations of $\mathbf{Q}$ and $\mathbf{Q}^{(\alpha,\beta)}$. Indeed,

$$\begin{aligned} \mathbf{Q} &\approx \mathbf{Q}_{\text{quad}} = \mathbf{U}^T[q]\mathbf{U}, \qquad \text{and} \\ \mathbf{Q}^{(\alpha,\beta)} &\approx \mathbf{Q}^{(\alpha,\beta)}_{\text{quad}} = \left(\mathbf{U}^{(\alpha)}\right)^T[q]\mathbf{U}^{(\beta)} = \left(\mathbf{D}^{(\alpha)}\right)^T \mathbf{Q}_{\text{quad}}\mathbf{D}^{(\beta)}, \end{aligned}$$

with $[q] = \operatorname{diag}_{\mathbf{k}\in\mathcal{K}}(q(\xi_{\mathbf{k}}))$. Next, we define the coordinate matrices

$$\mathbf{X}^{(\alpha)}_{\mathbf{j},\mathbf{k}} = (\varphi_{\mathbf{j}}, x_\alpha\varphi_{\mathbf{k}}), \qquad \mathbf{j},\mathbf{k} \in \mathcal{K}, \qquad \alpha = 1,\ldots,d.$$

Using $\mathbf{X}^{(\alpha)} = \mathbf{U}^T[x_\alpha]\mathbf{U}$ and the fact that $\mathbf{U}$ is an orthogonal matrix, one finds

$$\mathbf{Q}_{\text{quad}} = q(\mathbf{X}^{(1)},\ldots,\mathbf{X}^{(d)}) = q(\mathbf{X}), \tag{2.13}$$

where the right-hand side denotes formal insertion of the matrices $\mathbf{X}^{(\alpha)}$ into the multivariate polynomial q in place of x_α.

Turning back to the system (2.7), if we approximate $a^{(\alpha,\beta)}$ by polynomial interpolation, say $a^{(\alpha,\beta)} \approx a^{(\alpha,\beta)}_{\text{pol}}$, we can now compute the actions of the corresponding matrices according to

$$\mathbf{S}\mathbf{v} \approx \mathbf{S}_{\text{quad}}\mathbf{v} = \sum_{\alpha,\beta=1}^{d} \mathbf{S}^{(\alpha,\beta)}_{\text{quad}}\mathbf{v} = \sum_{\alpha,\beta=1}^{d} \left(\mathbf{D}^{(\alpha)}\right)^T a^{(\alpha,\beta)}_{\text{pol}}(\mathbf{X})\mathbf{D}^{(\beta)}\mathbf{v}. \tag{2.14}$$

The fast evaluation of these matrix-vector products is discussed in Section 3.

2.4. Treatment of boundary terms. First, we introduce some notation which makes the handling of the boundary terms more convenient. We denote the boundary faces by

$$\Omega^{(\pm\alpha)} = \{x \in \Omega \,:\, x_\alpha = \pm 1\}, \qquad \alpha = 1,\ldots,d.$$

We also introduce the notation

$$\Omega^{(\neg\alpha)} = \left\{x^{(\neg\alpha)} = (x_1,\ldots,x_{\alpha-1},x_{\alpha+1},\ldots,x_d) \,:\, x \in \Omega\right\},$$

where $x^{(\neg\alpha)}$ denotes the free coordinates on a boundary face and $\Omega^{(\neg\alpha)}$ denotes the dimensional reduction of the domain. Then, if f and g are separable functions, i.e., it holds that

ETNA
Kent State University
http://etna.math.kent.edu

$f(x) = f(x_\alpha) f_{\neg\alpha}(x^{(\neg\alpha)})$ and similarly for g, their L^2 inner product over a boundary face can be written as

$$(f, g)_{\Omega^{(\pm\alpha)}} = f_\alpha(\pm 1) g_\alpha(\pm 1) (f_{\neg\alpha}, g_{\neg\alpha})_{\Omega^{(\neg\alpha)}}.$$

Similarly, for a multi-index $\mathbf{k} \in \mathcal{K}$, we define

$$\mathbf{k}^{(\neg\alpha)} = (k_1, \ldots, k_{\alpha-1}, k_{\alpha+1}, \ldots, k_d).$$

Consider again an arbitrary polynomial $q\colon \Omega \to \mathbb{R}$ and its restrictions $q^{(\pm\alpha)} = q|_{\Omega^{(\pm\alpha)}}$ to the boundary faces. Note that $q^{(\pm\alpha)}$ only depends on the coordinates $x^{(\neg\alpha)}$. Using separability, the action of

$$\mathbf{Q}^{(\pm\alpha)}_{\mathbf{j},\mathbf{k}} = (\varphi_{\mathbf{j}}, q^{(\pm\alpha)} \varphi_{\mathbf{k}})_{\Omega^{(\pm\alpha)}}, \qquad \mathbf{j}, \mathbf{k} \in \mathcal{K},$$

on a vector $\mathbf{v} \in \mathbb{R}^{|\mathcal{K}|}$ is given by

$$\begin{aligned}\left(\mathbf{Q}^{(\pm\alpha)}\mathbf{v}\right)_{\mathbf{j}} &= \sum_{\mathbf{k}\in\mathcal{K}} \left(\varphi_{\mathbf{j}}, q^{(\pm\alpha)}\varphi_{\mathbf{k}}\right)_{\Omega^{(\pm\alpha)}} v_{\mathbf{k}} \\ &= \varphi_{j_\alpha}(\pm 1) \sum_{\mathbf{k}^{(\neg\alpha)}\in\mathcal{K}(d-1,K)} \left(\varphi_{\mathbf{j}^{(\neg\alpha)}}, q^{(\pm\alpha)}\varphi_{\mathbf{k}^{(\neg\alpha)}}\right)_{\Omega^{(\neg\alpha)}} \eta^{(\neg\alpha,\pm)}_{\mathbf{k}^{(\neg\alpha)}},\end{aligned}$$

with

$$\eta^{(\neg\alpha,\pm)}_{\mathbf{m}} = \sum_{m=0}^{K} \varphi_m(\pm 1) v_{(m_1,\ldots,m_{\alpha-1},m,m_\alpha,\ldots,m_{d-1})}, \qquad \mathbf{m} \in \mathcal{K}(d-1, K). \tag{2.15}$$

It takes $\mathcal{O}(|\mathcal{K}|)$ operations to compute $\eta^{(\neg\alpha,\pm)}$ from $\mathbf{v}$. Applying the same procedure as in the previous section, but now in $d-1$ dimensions, we can approximate

$$\left(\mathbf{Q}^{(\pm\alpha)}\mathbf{v}\right)_{\mathbf{j}} \approx \left(\mathbf{Q}^{(\pm\alpha)}_{\text{quad}}\mathbf{v}\right)_{\mathbf{j}} = \varphi_{j_\alpha}(\pm 1) \left(q^{(\pm\alpha)}\left(\mathbf{X}^{(\neg\alpha)}\right) \eta^{(\neg\alpha,\pm)}\right)_{\mathbf{j}^{(\neg\alpha)}}, \tag{2.16}$$

for all $\mathbf{j} \in \mathcal{K}$, where we set

$$q^{(\pm\alpha)}\left(\mathbf{X}^{(\neg\alpha)}\right) = q^{(\pm\alpha)}\left(\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(\alpha-1)}, \mathbf{X}^{(\alpha+1)}, \ldots, \mathbf{X}^{(d)}\right),$$

and $\mathbf{X}^{(\beta)}_{\mathbf{m},\mathbf{n}} = (\varphi_{\mathbf{m}}, x_\beta \varphi_{\mathbf{n}})$ with $\mathbf{m}, \mathbf{n} \in \mathcal{K}(d-1, K)$. Given $\eta^{(\neg\alpha,\pm)}$, the action of $q^{(\pm\alpha)}\left(\mathbf{X}^{(\neg\alpha)}\right)$ on a vector is computed efficiently using the procedure given in Section 3.1.

Turning back to the system (2.7), we define the restrictions $b^{(\pm\alpha)} = b|_{\Omega^{\pm(\alpha)}}$ and approximate $b^{(\pm\alpha)}$ by polynomial interpolation, say $b^{(\pm\alpha)} \approx b^{(\pm\alpha)}_{\text{pol}}$. The action of $\mathbf{B}$ on $\dot{\mathbf{v}}$ is then computable via

$$\mathbf{B}\dot{\mathbf{v}} = \sum_{\pm\alpha} \mathbf{B}^{(\pm\alpha)}\dot{\mathbf{v}} \approx \sum_{\pm\alpha} \mathbf{B}^{(\pm\alpha)}_{\text{quad}}\dot{\mathbf{v}}, \tag{2.17}$$

where $\mathbf{B}^{(\pm\alpha)}_{\mathbf{j},\mathbf{k}} = (\varphi_{\mathbf{j}}, b^{(\pm\alpha)}_{\text{pol}}\varphi_{\mathbf{k}})_{\Omega^{(\pm\alpha)}}$, and $\mathbf{B}^{(\pm\alpha)}_{\text{quad}}\dot{\mathbf{v}}$ is computed as in (2.16) with $b^{(\pm\alpha)}_{\text{pol}}$ in place of $q^{(\pm\alpha)}$. A fast evaluation procedure for matrix-vector products with $b^{(\pm\alpha)}_{\text{pol}}\left(\mathbf{X}^{(\neg\alpha)}\right)$ is given in Section 3.1.

ETNA
Kent State University
http://etna.math.kent.edu

2.5. Spatially discrete system and computational costs. To sum up, using (2.14) and (2.17) in (2.7), the system of ODEs we actually solve reads

$$\ddot{\mathbf{w}} = -\mathbf{S}_{\text{quad}}\mathbf{w} - \mathbf{B}_{\text{quad}}\dot{\mathbf{w}} + \mathbf{f} + \mathbf{g}. \tag{2.18}$$

We let this equation, together with appropriate initial conditions, define

$$w(x,t) = \sum_{\mathbf{k}\in\mathcal{K}} w_{\mathbf{k}}(t)\varphi_{\mathbf{k}}(x).$$

In the next section, we devise linearly scaling evaluation procedures for the right-hand side. More precisely, we show how the matrix-vector products (2.14) and (2.17) as occurring in (2.18) can be computed in

$$\mathcal{O}\left(d^2|\mathcal{R}(d,R)||\mathcal{K}(d,K)|\right) \quad \text{and} \quad \mathcal{O}\left(d|\mathcal{K}(d,K)| + d|\mathcal{R}(d-1,R)||\mathcal{K}(d-1,K)|\right)$$

operations, respectively, if all $a^{(\alpha,\beta)}$ are approximated using a Chebyshev polynomial tensor product expansion with polynomials indexed over a set $\mathcal{R}(d,R)$,

$$a^{(\alpha,\beta)}(x) \approx a^{(\alpha,\beta)}_{\text{pol}}(x) = \sum_{\mathbf{r}\in\mathcal{R}} \hat{a}^{(\alpha,\beta)}_{\mathbf{r}} \prod_{\gamma=1}^{d} T_{r_\gamma}(x_\gamma),$$

and if b is approximated analogously using an index set $\mathcal{R}(d-1,R)$.

For the wave equation written in first order form, cf. Section 6, stability for standard explicit time-stepping schemes imposes a time-step restriction $\Delta t \leq cK^{-2}$. This means that the total computational cost for solving the problem over a fixed time-interval scales as $\mathcal{O}(d^2K^2|\mathcal{R}||\mathcal{K}|)$.

3. Fast evaluation of matrix-vector products. In this section, we discuss how to efficiently compute the occurring matrix-vector products when propagating the spatially discretised system (2.18) in time. We consider general matrix-vector products of the forms

$$q(\mathbf{X})\mathbf{v}, \qquad \mathbf{D}^{(\alpha)}\mathbf{v}, \qquad \left(\mathbf{D}^{(\alpha)}\right)^T\mathbf{v},$$

as defined in (2.12) and (2.13), for an arbitrary vector $\mathbf{v}\in\mathbb{R}^{|\mathcal{K}|}$ and an arbitrary multivariate polynomial $q:\Omega\to\mathbb{R}$. The polynomial q serves as placeholder for the polynomially approximated coefficient functions occurring in (2.18). The matrix-vector product

$$\mathbf{Q}^{(\alpha,\beta)}_{\text{quad}}\mathbf{v} = \left(\mathbf{D}^{(\alpha)}\right)^T q(\mathbf{X})\mathbf{D}^{(\beta)}\mathbf{v}$$

can then be computed by performing the three above matrix-vector products sequentially, and the boundary terms $\mathbf{Q}^{(\pm\alpha)}_{\text{quad}}\mathbf{v}$ are computable via (2.16) and an analogous procedure for $q^{(\pm\alpha)}$ in $d-1$ dimensions. Section 3.1 contains the fast evaluation procedure for $q(\mathbf{X})\mathbf{v}$ which is built upon ideas from [4, 12], but deviates in two respects: First, it improves the computational costs. Second, it is adapted to a Legendre basis. If q is approximated in a tensor product basis of Chebyshev polynomials indexed over a set $\mathcal{R}(d,R)$, the computational cost for $q(\mathbf{X})\mathbf{v}$ scales as $|\mathcal{R}||\mathcal{K}|$. The products $\mathbf{D}^{(\alpha)}\mathbf{v}$ and $\left(\mathbf{D}^{(\alpha)}\right)^T\mathbf{v}$, whose computational costs scale as $|\mathcal{K}|$, are considered in Section 3.2.

ETNA
Kent State University
http://etna.math.kent.edu

3.1. Coefficient matrices. We start with the computation of $q(\mathbf{X})\mathbf{v}$. This is done using the fast algorithm from [4, 12], which we adapt here to a Legendre basis. We assume q to be a sum of separable polynomials. A general function can be approximated in this form using d-dimensional Chebyshev interpolation. Hence, we set

$$q(x) = \sum_{\mathbf{r}\in\mathcal{R}} \hat{q}_{\mathbf{r}} T_{\mathbf{r}}(x) = \sum_{\mathbf{r}\in\mathcal{R}} \hat{q}_{\mathbf{r}} \prod_{\alpha=1}^{d} T_{r_l}(x_l), \tag{3.1}$$

where each $\mathbf{r} = (r_1, \ldots, r_d)$ is a multi-index taken from a set $\mathcal{R}(d, R) \subset \mathbb{N}^d$, and T_{r_l} are univariate Chebyshev polynomials. When q is much smoother than the solution u, the corresponding index sets contain much fewer multi-indices than $\mathcal{K}(d, K)$. We then compute the matrix-vector product according to

$$q(\mathbf{X})\mathbf{v} = \sum_{\mathbf{r}\in\mathcal{R}} \hat{q}_{\mathbf{r}} \prod_{\alpha=1}^{d} T_{r_\alpha}(\mathbf{X}^{(\alpha)})\mathbf{v}. \tag{3.2}$$

The action of the one-dimensional (1D) coordinate matrix X, $X_{j,k} = (\varphi_j, x\varphi_k)$, can be computed in $\mathcal{O}(K)$ operations. This is done using the recurrence relation (2.5) and the orthogonality of the Legendre basis, viz.

$$\begin{aligned}(X\mathbf{v})_j &= \sum_{k=0}^{K} (\varphi_j, x\varphi_k) v_k \\ &= \sum_{k=0}^{K} \left(\frac{k+1}{\sqrt{(2k+1)(2k+3)}} \delta_{j,k+1} v_k + \frac{k}{\sqrt{(2k-1)(2k+1)}} \delta_{j,k-1} v_k \right) \\ &= \frac{j}{\sqrt{(2j-1)(2j+1)}} v_{j-1} + \frac{j+1}{\sqrt{(2j+1)(2j+3)}} v_{j+1}\end{aligned} \tag{3.3}$$

for all $j = 0, \ldots, K$, with $v_{-1} = v_{K+1} = 0$. Hence, if q is a degree R polynomial, $q(X)\mathbf{v}$ can be computed using (3.3) in combination with the Chebyshev recurrence relation

$$T_{r+1}(X)\mathbf{v} = 2XT_r(X)\mathbf{v} - T_{r-1}(X)\mathbf{v}, \quad T_1(X)\mathbf{v} = X\mathbf{v}, \quad T_0(X)\mathbf{v} = \mathbf{v}, \tag{3.4}$$

in $\mathcal{O}(RK)$ operations.

This procedure generalises to higher dimensions: Defining coordinate matrices for each dimension,

$$\mathbf{X}^{(\alpha)}_{\mathbf{j},\mathbf{k}} = (\varphi_{\mathbf{j}}, x_a \varphi_{\mathbf{k}}), \qquad \mathbf{j}, \mathbf{k} \in \mathcal{K}, \quad \alpha = 1, \ldots, d,$$

we find

$$\left(\mathbf{X}^{(\alpha)}\mathbf{v}\right)_{\mathbf{j}} = \frac{j_\alpha}{\sqrt{(2j_\alpha - 1)(2j_\alpha + 1)}} v_{\mathbf{j}-\mathbf{e}_\alpha} + \frac{j_\alpha + 1}{\sqrt{(2j_\alpha + 1)(2j_\alpha + 3)}} v_{\mathbf{j}+\mathbf{e}_\alpha} \tag{3.5}$$

for all $\mathbf{j} \in \mathcal{K}$, where $\mathbf{e}_\alpha$ is the αth unit vector, $v_{\mathbf{j}-\mathbf{e}_\alpha} = 0$ if $j_\alpha = 0$, and $v_{\mathbf{j}+\mathbf{e}_\alpha} = 0$ if $j_\alpha = K$. Thus, $\mathbf{X}^{(\alpha)}\mathbf{v}$ is obtained in $\mathcal{O}(|\mathcal{K}|)$ operations. We introduce the notation

$$\mathbf{j} \overset{\alpha}{\leftarrow} j = (j_0, \ldots, j_{\alpha-1}, j, j_{\alpha+1}, \ldots, j_d), \qquad \mathbf{j} = (j_1, \ldots, j_d) \in \mathbb{N}^d.$$

The idea, which slightly modifies and improves the procedure given in [4], is to compute the right-hand side of (3.2) term-wise starting from $\alpha = 1$ and $\mathbf{r} = \mathbf{0}$. For each choice of

ETNA
Kent State University
http://etna.math.kent.edu

Algorithm 1: Recursive procedure for $\eta = q(\mathbf{X})\mathbf{v}$ starting from given $\mathbf{v}$, $\eta = \mathbf{0}$, $\alpha = 1$, and $\mathbf{r} = (0, \ldots, 0) \in \mathbb{N}^d$, where q is given as in (3.1).

```
function η = fastalgorithm (v, η, α, r)
for r = 0 to R do
    if r = 0 then
        v_- := v
    else if r = 1 then
        v_+ := X^(α) v                                   use (3.5): O(|K|)
    else
        temp := v_+
        v_+ := 2X^(α) v_+ − v_-                          use (3.4) together with (3.5): O(|K|)
        v_- := temp
    w := { v_-,  r = 0,
         { v_+,  else,
    r := r ←^α r
    if α < d then
        η := fastalgorithm(w, η, α+1, r)                 recursion: next coordinate
    else
        η := η + q̂_r w                                   last coordinate: sum up
```

r, starting from $r = 0$, we carry out a single step of the 1D Chebyshev recurrence (3.4) in combination with (3.5) and then carry on recursively with $\alpha + 1$ and $\mathbf{r} \xleftarrow{\alpha} r$. Having reached $\alpha = d$, we multiply the computed vector by $\hat{q}_{\mathbf{r}}$, where $\mathbf{r}$ is the currently considered term in (3.2), and sum up the result. The matrix-vector product $\eta = q(\mathbf{X})\mathbf{v}$ can thus be computed in $\mathcal{O}(|\mathcal{R}||\mathcal{K}|)$ operations. A pseudocode formulation of this recursive procedure is given in Algorithm 1.

3.2. Differentiation matrices. Matrix-vector products with the 1D differentiation matrix D, $D_{j,k} = (\varphi_j, \varphi_k')$, can be evaluated recursively as

$$(D\mathbf{v})_{j-2} = \sqrt{\frac{2j-3}{2j+1}}(D\mathbf{v})_j + \sqrt{(2j-1)(2j-3)}v_{j-1}, \qquad j = 2, \ldots, K,$$
$$(D\mathbf{v})_K = 0, \qquad (D\mathbf{v})_{K-1} = \sqrt{(2K+1)(2K-1)}v_K,$$

see, e.g., [5, Ch. 2.3]. Clearly, the computational work scales linearly with K. Similarly, the product $D^T\mathbf{v}$ can be evaluated as

$$(D^T\mathbf{v})_j = \sqrt{\frac{2j+1}{2j-3}}(D^T\mathbf{v})_{j-2} + \sqrt{(2j+1)(2j-1)}v_{j-1}, \quad j = 2, \ldots, K, \tag{3.6}$$
$$(D^T\mathbf{v})_0 = 0, \qquad (D^T\mathbf{v})_1 = \sqrt{3}v_0.$$

In higher dimensions, we wish to evaluate the matrix-vector product

$$\mathbf{D}^{(\alpha)}\mathbf{v} = \left(\sum_{\mathbf{k}\in\mathcal{K}}(\varphi_{\mathbf{j}}, \frac{\partial}{\partial x_\alpha}\varphi_{\mathbf{k}})v_{\mathbf{k}}\right)_{\mathbf{j}\in\mathcal{K}},$$

ETNA
Kent State University
http://etna.math.kent.edu

as well as $(\mathbf{D}^{(\alpha)})^T\mathbf{v}$. Since the basis is orthonormal,

$$\left(\mathbf{D}^{(\alpha)}\mathbf{v}\right)_{\mathbf{j}} = \sum_{k_\alpha=0}^{K} (\varphi_{j_\alpha}, \varphi'_{k_\alpha}) v_{\mathbf{j}\overset{\alpha}{\leftarrow}k_\alpha}, \qquad \mathbf{j} \in \mathcal{K}.$$

Thus, the computation of $\mathbf{D}^{(\alpha)}\mathbf{v}$ boils down to computing the action of the 1D differentiation matrix D recursively. This needs to be done for all indices $\mathbf{j}^{(\neg\alpha)}$. The matrix-vector product $(\mathbf{D}^{(\alpha)})^T\mathbf{v}$ is computed analogously using (3.6). Both algorithms scale linearly with $|\mathcal{K}|$.

4. Well-posedness and stability. In this section, we briefly derive energy estimates to prove well-posedness of the continuous problem and stability of its spatial semidiscretisation. A problem is strongly well-posed if it has a unique solution satisfying an energy estimate of the form

$$\|u(\cdot,t)\|_E^2 \le \kappa(T)\left(\|u(\cdot,0)\|_E^2 + \int_0^t (\|f(\cdot,\tau)\|^2 + \|g(\cdot,\tau)\|_{\partial\Omega}^2)\,\mathrm{d}\tau\right) \quad \forall t \in [0,T], \tag{4.1}$$

where κ is independent of u_1, u_2, f and g, and is bounded for any finite T [17]. Strong well-posedness of the problem (2.1)–(2.4) is given by Lemma 4.1 below. The spatial semidiscretisation given in Section 2 is constructed such that the resulting system of ODEs satisfies a similar energy estimate as the PDE. A semidiscretisation with such a property is called (strongly) stable [17].

LEMMA 4.1. *The problem* (2.1)–(2.4) *is strongly well-posed in the seminorm*

$$\|u\|_E = \left(\|u_t\|^2 + (\nabla u, a\nabla u)\right)^{1/2}. \tag{4.2}$$

Proof. We use the energy method [17]. First, differentiate $\|u(\cdot,t)\|_E^2$ with respect to t. With $b(x) \ge b_0 > 0$, the contribution from the boundary forcing can be bounded by completing the square. By integrating the resulting differential inequality, we get (4.2) with $\kappa = \mathrm{e}\max(1, 1/2b_0, T)$ for all $t \in [0,T]$. ∎

Strong stability of the semidiscretisation (2.18) with respect to a discrete analogue of the seminorm (4.2) can be established using similar arguments.

LEMMA 4.2. *Provided* $\det(a_{\text{pol}}(x)) \ge a_0 > 0$ *and* $b_{\text{pol}}(x) \ge b_0 > 0$ *for all* x, *the semidiscretisation* (2.18) *is strongly stable in the discrete seminorm*

$$\|\mathbf{w}\|_{\mathbf{E}}^2 = \|\dot{\mathbf{w}}\|^2 + \mathbf{w}^T\mathbf{S}_{\text{quad}}\mathbf{w}. \tag{4.3}$$

Proof. If $\det(a_{\text{pol}}(x)) \ge a_0 > 0$, the inner matrix in

$$\mathbf{S}_{\text{quad}} = \begin{bmatrix}\mathbf{U}^{(1)}\\ \vdots \\ \mathbf{U}^{(d)}\end{bmatrix}^T \begin{bmatrix}[a_{\text{pol}}^{(1,1)}] & \cdots & [a_{\text{pol}}^{(1,d)}]\\ \vdots & \ddots & \vdots \\ [a_{\text{pol}}^{(d,1)}] & \cdots & [a_{\text{pol}}^{(d,d)}]\end{bmatrix} \begin{bmatrix}\mathbf{U}^{(1)}\\ \vdots \\ \mathbf{U}^{(d)}\end{bmatrix} \tag{4.4}$$

is positive definite. Then, $\mathbf{S}_{\text{quad}}$ is positive semidefinite, which implies that $\|\cdot\|_{\mathbf{E}}$ is a seminorm. If $b_{\text{pol}}(x) \ge b_0 > 0$, $\mathbf{B}$ is positive semidefinite, and positive definite with respect to the boundary norm. We now prove that this property transfers to $\mathbf{B}_{\text{quad}}$. Considering the

ETNA
Kent State University
http://etna.math.kent.edu

boundary face $\Omega^{(\pm\alpha)}$, for any $v = \sum_{\mathbf{k}\in\mathcal{K}} v_{\mathbf{k}}\varphi_{\mathbf{k}}$ it holds that

$$\begin{aligned}
\mathbf{v}^T\mathbf{B}_{\text{quad}}^{(\pm\alpha)}\mathbf{v} &= \sum_{\mathbf{k}\in\mathcal{K}} v_{\mathbf{k}}(\mathbf{B}_{\text{quad}}^{(\pm\alpha)}\mathbf{v})_{\mathbf{k}} \\
&= \sum_{\mathbf{k}\in\mathcal{K}} v_{\mathbf{k}}\varphi_{k_\alpha}(\pm 1)(b_{\text{pol}}^{(\pm\alpha)}(\mathbf{X}^{(\neg\alpha)})\eta^{(\neg\alpha,\pm)})_{\mathbf{k}^{(\neg\alpha)}} \\
&= (\eta^{(\neg\alpha,\pm)})^T b_{\text{pol}}^{(\pm\alpha)}(\mathbf{X}^{(\neg\alpha)})\eta^{(\neg\alpha,\pm)} \\
&= \sum_{\mathbf{k}\in\mathcal{K}(d-1,K)} \omega_{\mathbf{k}} b_{\text{pol}}^{(\pm\alpha)}(\xi_{\mathbf{k}}) \left(\sum_{\mathbf{j}\in\mathcal{K}(d-1,K)} \eta_{\mathbf{j}}^{(\neg\alpha,\pm)}\varphi_{\mathbf{j}}(\xi_{\mathbf{k}}) \right)^2 \\
&\geq b_0\|v\|^2_{\Omega^{(\pm\alpha)}}.
\end{aligned}$$

With these properties established, if we multiply (2.18) by $\dot{\mathbf{w}}^T$ from the left the result follows using the same reasoning as in Lemma 4.1. We get the estimate

$$\|\mathbf{w}(t)\|^2_{\mathbf{E}} \leq \kappa(T)\left(\|\mathbf{w}(0)\|^2_{\mathbf{E}} + \int_0^t (\|f(\cdot,\tau)\|^2 + \|g(\cdot,\tau)\|^2_{\partial\Omega})\,\mathrm{d}\tau\right) \qquad \forall\, t \in [0,T],$$

with $\kappa = \mathrm{e}\max(1, 1/2b_0, T)$, cf. (4.1). ∎

5. Error analysis. In this section, C and κ denote general constants, which may adopt different values at different occurrences. We let $\|\cdot\|_{H^s(\Omega)}$ and $|\cdot|_{H^s(\Omega)}$ denote the standard Sobolev norms and seminorms, respectively.

We let u denote the solution of (2.1)–(2.4). The coefficient functions a and b are approximated by degree R polynomial interpolants a_{pol} and b_{pol}, respectively, cf. Section 2.5. For the Legendre–Galerkin approximation of u, we use an index set $\mathcal{K}(d,K)$, and we let $w = \sum_{\mathbf{k}\in\mathcal{K}} w_{\mathbf{k}}\varphi_{\mathbf{k}} \in \mathcal{V}_{\mathcal{K}}$ be the solution of the semidiscretised problem (2.18), where also the effects of quadrature are taken into account.

We show that the method devised in this paper is spectrally accurate with respect to the energy seminorm (4.2). It gives rise to three different errors. First, there is an error due to polynomial interpolation of the coefficient functions, which can be bounded in terms of the interpolation errors and of the exact solution using energy estimates. This is done in Section 5.1 following standard theory. Second, we truncate the basis. Third, the matrix-free method is equivalent to a quadrature approximation, the resulting errors are studied in Section 5.2, where new techniques come into play. They are bounded in terms of the exact solution using energy estimates in combination with a standard Legendre approximation result and an appropriate projection which is related to the exactness of Gaussian quadrature. Putting these results together then yields the following error bound.

For matrix-valued arguments, the Sobolev norm is understood as the Frobenius norm of the Sobolev norms of the matrix elements.

THEOREM 5.1. *Assume that $u(\cdot,t) \in H^s(\Omega)$ and $u_t(\cdot,t) \in H^{s-1}(\Omega)$ for all $t \in [0,T]$, with $s \geq 4$. Assume also that the degree R interpolants a_{pol} and b_{pol} of a and b satisfy*

$$\|a - a_{\text{pol}}\|_{H^1(\Omega)} \leq \varepsilon, \qquad \|b - b_{\text{pol}}\|_{L^2(\partial\Omega)} \leq \varepsilon,$$

with $R \leq K/2$. Then, the error of the solution $w \in \mathcal{V}_{\mathcal{K}(d,K)}$ of the semidiscretised problem (2.18) is bounded by

$$\|u(\cdot,t) - w(\cdot,t)\|^2_E \leq C\left(C_2\varepsilon^2 + C_s K^{8-2s}\right) \qquad \forall\, t \in [0,T],$$

ETNA
Kent State University
http://etna.math.kent.edu

where $C = C(d, a, b, \Omega, T)$ is independent of u, K, and R, and depends linearly on T, and where

$$C_r = \int_0^t \left(\|u(\cdot,\tau)\|^2_{H^r(\Omega)} + \|u_t(\cdot,\tau)\|^2_{H^{r-1}(\Omega)} \right) \mathrm{d}\tau,$$

with $r = 2, s$.

5.1. Interpolation error. We first estimate the error due to approximation of the coefficients a and b by their Chebyshev interpolants a_{pol} and b_{pol}, using the energy method.

LEMMA 5.2. *If $u(\cdot,t) \in H^2(\Omega)$ is the solution at time t of (2.1)–(2.4), $u_t(\cdot,t) \in H^1(\Omega)$, and if $u^{\text{pol}}(\cdot,t) \in H^1(\Omega)$ is the solution at time t of the same problem with the coefficients a and b exchanged with their Chebyshev interpolants, the error due to interpolation is given by*

$$\|u(\cdot,t) - u^{\text{pol}}(\cdot,t)\|^2_E \le C \int_0^t \Big(\|a - a_{\text{pol}}\|^2_{H^1(\Omega)} \|u(\cdot,\tau)\|^2_{H^2(\Omega)} + \|b - b_{\text{pol}}\|^2_{L^2(\partial\Omega)} \|u_t(\cdot,\tau)\|^2_{H^1(\Omega)} \Big) \mathrm{d}\tau$$

for all $t \in [0,T]$, where $C = C(\Omega, T)$ is independent of u and depends linearly on T.

Proof. u and u^{pol} satisfy the weakly formulated problems

$$\begin{aligned} (\phi, u_{tt}) &= -(\nabla\phi, a\nabla u) - (\phi, bu_t)_{\partial\Omega} + (\phi, f) + (\phi, g)_{\partial\Omega} \quad \forall \phi \in H^1(\Omega), \\ (\phi, u^{\text{pol}}_{tt}) &= -(\nabla\phi, a_{\text{pol}}\nabla u^{\text{pol}}) - (\phi, b_{\text{pol}} u^{\text{pol}}_t)_{\partial\Omega} + (\phi, f) + (\phi, g)_{\partial\Omega} \quad \forall \phi \in H^1(\Omega), \end{aligned} \tag{5.1}$$

with the initial data $u(x,0) = u^{\text{pol}}(x,0) = u_1(x)$, and $u_t(x,0) = u^{\text{pol}}_t(x,0) = u_2(x)$. By subtracting the equations, we see that the error $e^{\text{pol}} = u - u^{\text{pol}}$ satisfies

$$(\phi, e^{\text{pol}}_{tt}) = -(\nabla\phi, a_{\text{pol}}\nabla e^{\text{pol}}) - (\phi, b_{\text{pol}} e^{\text{pol}}_t)_{\partial\Omega} + (\phi, f^{\text{pol}}) + (\phi, g^{\text{pol}})_{\partial\Omega} \tag{5.2}$$

$\forall \phi \in H^1(\Omega)$, with $e^{\text{pol}}(x,0) = e^{\text{pol}}_t(x,0) = 0$, and

$$f^{\text{pol}} = \nabla \cdot (a - a_{\text{pol}})\nabla u, \qquad g^{\text{pol}} = -\mathbf{n} \cdot (a - a_{\text{pol}})\nabla u - (b - b_{\text{pol}})u_t.$$

Since (5.2) is of the same form as (5.1), it satisfies a similar energy estimate. We can thereby bound e^{pol} in terms of the interpolation errors $a - a_{\text{pol}}$ and $b - b_{\text{pol}}$, and of the exact solution u. The proof is completed by manipulation of Sobolev norms and use of the trace inequality. ∎

The interpolation errors $a - a_{\text{pol}}$ and $b - b_{\text{pol}}$ can be bounded in terms of the order R of the interpolants and of the derivatives of a and b using standard theory, see [5]. For smooth coefficients, the error decays faster than any polynomial in R.

5.2. Spatial discretisation and quadrature. In this section, we study the error when comparing the exact solution to the semidiscretisation, approximation of the matrix elements by quadrature taken into account. Throughout this section, to keep the notation clean, we drop the subscript $(\cdot)_{\text{pol}}$ and let a and b refer to the polynomial interpolants of the coefficients.

Let $\mathcal{P}_\mathcal{K}$ and $\mathcal{P}^\perp_\mathcal{K}$ be the L^2-orthogonal projection onto the polynomial approximation space $\mathcal{V}_\mathcal{K}$ and its orthogonal complement, respectively. The exact solution

$$u(x,t) = \sum_{\mathbf{k} \in \mathbb{N}^d} u_\mathbf{k}(t) \varphi_\mathbf{k}(x)$$

ETNA
Kent State University
http://etna.math.kent.edu

of (2.1)–(2.4) satisfies, in particular,

$$(\varphi_{\mathbf{j}}, \mathcal{P}_{\mathcal{K}} u_{tt}) = -(\nabla\varphi_{\mathbf{j}}, a\nabla\mathcal{P}_{\mathcal{K}} u) - (\varphi_{\mathbf{j}}, b\mathcal{P}_{\mathcal{K}} u_t)_{\partial\Omega} + (\varphi_{\mathbf{j}}, f + f_{\mathcal{K}}) + (\varphi_{\mathbf{j}}, g + g_{\mathcal{K}})_{\partial\Omega}$$

for all $\mathbf{j} \in \mathcal{K}$, with the internal and boundary defects

$$f_{\mathcal{K}} = \nabla \cdot a\nabla\mathcal{P}_{\mathcal{K}}^{\perp} u, \qquad g_{\mathcal{K}} = -\mathbf{n} \cdot a\nabla\mathcal{P}_{\mathcal{K}}^{\perp} u - b\mathcal{P}_{\mathcal{K}}^{\perp} u_t.$$

If we let $\mathbf{u} = (u_{\mathbf{k}})_{\mathbf{k}\in\mathcal{K}}$, $\mathbf{f}_{\mathcal{K}} = (f_{\mathcal{K},\mathbf{j}})_{\mathbf{j}\in\mathcal{K}}$, $\mathbf{g}_{\mathcal{K}} = (g_{\mathcal{K},\mathbf{j}})_{\mathbf{j}\in\mathcal{K}}$, where

$$f_{\mathcal{K},\mathbf{j}} = (\varphi_{\mathbf{j}}, f_{\mathcal{K}}), \qquad g_{\mathcal{K},\mathbf{j}} = (\varphi_{\mathbf{j}}, g_{\mathcal{K}})_{\partial\Omega},$$

this can equivalently be written as

$$\ddot{\mathbf{u}} = -\mathbf{S}\mathbf{u} - \mathbf{B}\dot{\mathbf{u}} + \mathbf{f} + \mathbf{g} + \mathbf{f}_{\mathcal{K}} + \mathbf{g}_{\mathcal{K}}. \tag{5.3}$$

Similarly, let $\mathbf{w} = (w_{\mathbf{k}})_{\mathbf{k}\in\mathcal{K}}$ be the solution of the semidiscrete system (2.18). Following [34], we decompose the error according to

$$u - \sum_{\mathbf{k}\in\mathcal{K}} w_{\mathbf{k}}\varphi_{\mathbf{k}} = (u - \mathcal{P}_{\mathcal{K}} u) + \left(\mathcal{P}_{\mathcal{K}} u - \sum_{\mathbf{k}\in\mathcal{K}} w_{\mathbf{k}}\varphi_{\mathbf{k}}\right) =: \rho + \theta,$$

where ρ can be bounded using the standard projection estimate [5]

$$\|u - \mathcal{P}_{\mathcal{K}} u\|_{H^n(\Omega)} \le CK^{\nu-s}|u|_{H^s(\Omega)}, \qquad \nu = \begin{cases} 0, & n = 0, \\ 2n - \frac{1}{2}, & n \ge 1. \end{cases} \tag{5.4}$$

The error θ or, equivalently, $\mathbf{u} - \mathbf{w}$, is bounded in Lemma 5.3 below. The proof is based on standard energy estimates, but to bound the effect of quadrature error we need to introduce an appropriate projection matrix which is related to the exactness properties of Gaussian quadrature.

LEMMA 5.3. *Let $u(\cdot, t)$ be the solution at time t of (2.1)–(2.4) with the coefficient functions replaced with their degree R polynomial interpolants, and $\mathcal{P}_{\mathcal{K}} u(\cdot, t) = \sum_{\boldsymbol{k}\in\mathcal{K}} u_{\boldsymbol{k}}(t)\varphi_{\boldsymbol{k}}$ be the orthogonal projection of $u(\cdot, t)$ onto the polynomial approximation space $\mathcal{V}_{\mathcal{K}}$, with $\mathcal{K} = \mathcal{K}(d, K)$. Denote $\boldsymbol{u}(t) = (u_{\boldsymbol{k}}(t))_{\boldsymbol{k}\in\mathcal{K}}$, and let $\mathbf{w}(t)$ be the solution at time t of the semidiscrete approximation (2.18). Then,*

$$\|\boldsymbol{u}(t) - \mathbf{w}(t)\|_{\boldsymbol{E}}^2 \le C(K - R)^{-2s} \int_0^t \left(K^8 |u(\cdot, \tau)|_{H^s(\Omega)}^2 + (K - R)^5 |u_t(\cdot, \tau)|_{H^{s-1}(\Omega)}^2\right) \mathrm{d}\tau$$

for all $t \in [0, T]$, where $C = C(d, a, b, \Omega, T)$ is independent of u, K, and R, and depends linearly on T.

Proof. We set $\mathbf{e} = \mathbf{u} - \mathbf{w}$ and subtract (2.18) from (5.3) to obtain

$$\ddot{\mathbf{e}} = -\mathbf{S}_{\text{quad}}\mathbf{e} - \mathbf{B}_{\text{quad}}\dot{\mathbf{e}} + \mathbf{f}^{\text{quad}} + \mathbf{f}_{\mathcal{K}} + \mathbf{g}^{\text{quad}} + \mathbf{g}_{\mathcal{K}},$$

where

$$\mathbf{f}^{\text{quad}} = -(\mathbf{S} - \mathbf{S}_{\text{quad}})\mathbf{u}, \qquad \mathbf{g}^{\text{quad}} = -(\mathbf{B} - \mathbf{B}_{\text{quad}})\dot{\mathbf{u}}.$$

We can bound the error $\mathbf{e}$ using energy estimates of the same form as before. Differentiating the discrete seminorm (4.3) of $\mathbf{e}$ with respect to time yields

$$\begin{aligned} \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\mathbf{e}\|_{\mathbf{E}}^2 &= \dot{\mathbf{e}}^T(\mathbf{f}^{\text{quad}} + \mathbf{f}_{\mathcal{K}}) - \dot{\mathbf{e}}^T\mathbf{B}_{\text{quad}}\dot{\mathbf{e}} + \dot{\mathbf{e}}^T(\mathbf{g}^{\text{quad}} + \mathbf{g}_{\mathcal{K}}), \\ \frac{\mathrm{d}}{\mathrm{d}t}\|\mathbf{e}\|_{\mathbf{E}}^2 &\le \frac{1}{T}\|\dot{\mathbf{e}}\|^2 + T\|\mathbf{f}^{\text{quad}} + \mathbf{f}_{\mathcal{K}}\|^2 - 2\dot{\mathbf{e}}^T\mathbf{B}_{\text{quad}}\dot{\mathbf{e}} + 2\dot{\mathbf{e}}^T(\mathbf{g}^{\text{quad}} + \mathbf{g}_{\mathcal{K}}) \\ &\le \frac{1}{T}\|\mathbf{e}\|_{\mathbf{E}}^2 + 2T\|\mathbf{f}^{\text{quad}}\|^2 + 2T\|\mathbf{f}_{\mathcal{K}}\|^2 - 2\dot{\mathbf{e}}^T\mathbf{B}_{\text{quad}}\dot{\mathbf{e}} + 2\dot{\mathbf{e}}^T(\mathbf{g}^{\text{quad}} + \mathbf{g}_{\mathcal{K}}) \end{aligned} \tag{5.5}$$

ETNA
Kent State University
http://etna.math.kent.edu

for all $t \in [0, T]$. We first treat the contributions from the internal forcings $\mathbf{f}^{\text{quad}}$ and $\mathbf{f}_{\mathcal{K}}$, and then turn to the boundary forcings $\mathbf{g}^{\text{quad}}$ and $\mathbf{g}_{\mathcal{K}}$.

Most elements of $\mathbf{S} - \mathbf{S}_{\text{quad}}$ and $\mathbf{B} - \mathbf{B}_{\text{quad}}$ are zero. We get a non-vanishing quadrature error only in the elements $(\mathbf{j}, \mathbf{k})$ with $|\mathbf{j} + \mathbf{k}|_\infty = \max(\mathbf{j} + \mathbf{k}) > 2K + 1 - R$. Define the diagonal projection matrix $\mathbf{P}$ as $\mathbf{P}_{\mathbf{j},\mathbf{j}} = 1$ for $|\mathbf{j}|_\infty > K + 1 - R$, and $\mathbf{P}_{\mathbf{j},\mathbf{j}} = 0$ otherwise. Then,

$$\mathbf{S} - \mathbf{S}_{\text{quad}} = \mathbf{P}(\mathbf{S} - \mathbf{S}_{\text{quad}})\mathbf{P},$$

and

$$\|\mathbf{f}^{\text{quad}}\| = \|(\mathbf{S} - \mathbf{S}^{\text{quad}})\mathbf{u}\| \leq (\|\mathbf{S}\| + \|\mathbf{S}^{\text{quad}}\|)\|\mathbf{P}\mathbf{u}\|, \tag{5.6}$$

with the spectral matrix norm

$$\|A\| = \sup_{\|\mathbf{v}\|=1} \|A\mathbf{v}\|.$$

The formal extension of $\mathbf{P}$ to an infinite matrix, with ones on the new diagonal elements, can be interpreted as the matrix representation of an operator $\mathcal{P} : L^2 \to L^2$. With this operator, we can rewrite $\mathbf{P}\mathbf{u}$ to get

$$\|\mathbf{P}\mathbf{u}\| = \|\mathcal{P}\mathcal{P}_{\mathcal{K}}u\| = \|\mathcal{P}u - \mathcal{P}\mathcal{P}_{\mathcal{K}}^{\perp}u\| \leq \|\mathcal{P}u\| + \|\mathcal{P}_{\mathcal{K}}^{\perp}u\|,$$

and, by the projection estimate (5.4),

$$\|\mathbf{P}\mathbf{u}\| \leq C(K - R)^{-s}|u|_{H^s(\Omega)}.$$

This bound is conservative, but still spectrally accurate with respect to K for $K \gg R$. Using (4.4), the approximate stiffness matrix can be bounded by

$$\|\mathbf{S}_{\text{quad}}\| \leq \sum_{\alpha,\beta=1}^{d} a_{\max}\|\mathbf{U}^{(\alpha)}\|\|\mathbf{U}^{(\beta)}\|.$$

Using the exactness of Gaussian quadrature, we compute

$$\|\mathbf{U}^{(\alpha)}\mathbf{v}\|^2 = \sum_{\mathbf{l}\in\mathcal{K}} \omega_{\mathbf{l}} \left(\sum_{\mathbf{j}\in\mathcal{K}} v_{\mathbf{j}} \frac{\partial}{\partial x_\alpha} \varphi_{\mathbf{j}}(\xi_{\mathbf{l}}) \right)^2 = \int_\Omega \left(\sum_{\mathbf{j}\in\mathcal{K}} v_{\mathbf{j}} \frac{\partial}{\partial x_\alpha} \varphi_{\mathbf{j}}(x) \right)^2 \mathrm{d}x.$$

One can show that $\|\varphi_k'\| \leq Ck^{3/2}$. As $\mathbf{U}^{(\alpha)}$ is the Kronecker product of its one-dimensional counterpart and $d - 1$ orthogonal matrices, its norm is independent of d. Using that and the triangle and Cauchy–Schwarz inequalities,

$$\begin{aligned}\|\mathbf{U}^{(\alpha)}\| &= \sup_{\|\mathbf{v}\|=1} \|\mathbf{U}^{(\alpha)}\mathbf{v}\| = \sup_{\|\mathbf{v}\|=1} \left\| \sum_{k=1}^{K} v_k \varphi_k' \right\| \\ &\leq \sup_{\|\mathbf{v}\|=1} \sum_{k=0}^{K} |v_k|\|\varphi_k'\| \leq \sup_{\|\mathbf{v}\|=1} C\|\mathbf{v}\| \sqrt{\sum_{k=0}^{K} k^3} \leq CK^2.\end{aligned}$$

Consequently,

$$\|\mathbf{S}_{\text{quad}}\| \leq Cd^2 a_{\max} K^4. \tag{5.7}$$

ETNA
Kent State University
http://etna.math.kent.edu

We derive a similar estimate for $\|\mathbf{S}\|$. Since $\mathbf{S}$ is symmetric, we can estimate its norm using Rayleigh quotients,

$$\begin{aligned}\|\mathbf{S}\| &= \sup_{\|\mathbf{v}\|=1} \frac{\mathbf{v}^T\mathbf{S}\mathbf{v}}{\mathbf{v}^T\mathbf{v}} = \sup_{\|\mathbf{v}\|=1} \sum_{\mathbf{j},\mathbf{k}\in\mathcal{K}} v_\mathbf{j}\mathbf{S}_{\mathbf{j},\mathbf{k}}v_\mathbf{k} = \sup_{\|\mathbf{v}\|=1} \left(\left(\sum_{\mathbf{j}\in\mathcal{K}} v_\mathbf{j}\nabla\varphi_\mathbf{j}\right), a\left(\sum_{\mathbf{j}\in\mathcal{K}} v_\mathbf{j}\nabla\varphi_\mathbf{j}\right)\right) \\ &\le \sup_{\|\mathbf{v}\|=1} a_{\max}\left\|\sum_{\mathbf{j}\in\mathcal{K}} v_\mathbf{j}\nabla\varphi_\mathbf{j}\right\|^2 \le Cda_{\max}K^4.\end{aligned}$$

Together with (5.7) and (5.6), this yields

$$\|\mathbf{f}^{\text{quad}}\| \le Cd^2a_{\max}K^4(K-R)^{-s}|u|_{H^s(\Omega)}.$$

The norm of $\mathbf{f}_\mathcal{K}$ is bounded by

$$\begin{aligned}\|\mathbf{f}_\mathcal{K}\| \le \|f_\mathcal{K}\| &= \|\nabla\cdot a\nabla\mathcal{P}_\mathcal{K}^\perp u\| \le |a|_{H^1(\Omega)}|\mathcal{P}_\mathcal{K}^\perp u|_{H^1(\Omega)} + \|a\||\mathcal{P}_\mathcal{K}^\perp u|_{H^2(\Omega)} \\ &\le CK^{7/2-s}|u|_{H^s(\Omega)}.\end{aligned}$$

Bounding the influence of the boundary forcings $\mathbf{g}^{\text{quad}}$ and $\mathbf{g}_\mathcal{K}$ is done using similar ideas, but we need to work on the boundary faces—it is not enough to bound $\|\mathbf{g}^{\text{quad}}+\mathbf{g}_\mathcal{K}\|$. Eq. (2.15), which extracts the Legendre coefficients on a boundary face, is a linear transformation from $\mathbb{R}^{|\mathcal{K}(d,K)|}$ to $\mathbb{R}^{|\mathcal{K}(d-1,K)|}$. It can therefore be written as

$$\eta^{(\neg\alpha,\pm)} = \mathbf{H}^{(\pm\alpha)}\mathbf{u},$$

where the matrix $\mathbf{H}^{(\pm\alpha)}$ is defined implicitly through (2.15). This allows us to write $\mathbf{B}^{(\pm\alpha)}$ and $\mathbf{B}_{\text{quad}}^{(\pm\alpha)}$ as

$$\mathbf{B}^{(\pm\alpha)} = (\mathbf{H}^{(\pm\alpha)})^T\widehat{\mathbf{B}}^{(\pm\alpha)}\mathbf{H}^{(\pm\alpha)}, \qquad \mathbf{B}_{\text{quad}}^{(\pm\alpha)} = (\mathbf{H}^{(\pm\alpha)})^T\widehat{\mathbf{B}}_{\text{quad}}^{(\pm\alpha)}\mathbf{H}^{(\pm\alpha)},$$

with the $|\mathcal{K}(d-1,K)|\times|\mathcal{K}(d-1,K)|$-matrices

$$\widehat{\mathbf{B}}_{\text{quad}}^{(\pm\alpha)} = \mathbf{U}^T[b^{(\pm\alpha)}]\mathbf{U}, \qquad (\widehat{\mathbf{B}}^{(\pm\alpha)})_{\mathbf{j},\mathbf{k}} = (\varphi_\mathbf{j}, b^{(\pm\alpha)}\varphi_\mathbf{k})_{\Omega^{(\pm\alpha)}}, \quad \mathbf{j},\mathbf{k}\in\mathcal{K}(d-1,K),$$

where $\mathbf{U}$ is defined as in (2.10), but in $d-1$ dimensions. We decompose the boundary forcings $\mathbf{g}^{\text{quad}} = \sum \mathbf{g}^{(\pm\alpha),\text{quad}}$ and $\mathbf{g}_\mathcal{K} = \sum \mathbf{g}_\mathcal{K}^{(\pm\alpha)}$ into their contributions on each boundary face. We also note that for any $\mathbf{v}\in\mathbb{R}^{|\mathcal{K}|}$,

$$\mathbf{v}^T\mathbf{g}_\mathcal{K}^{(\pm\alpha)} = \mathbf{v}^T(\mathbf{H}^{(\pm\alpha)})^T\widehat{\mathbf{g}}_\mathcal{K}^{(\pm\alpha)}, \qquad \widehat{g}_{\mathcal{K},\mathbf{j}}^{(\pm\alpha)} = (\varphi_\mathbf{j}, g_\mathcal{K}|_{x_\alpha=\pm1})_{\Omega^{(\neg\alpha)}}, \quad \mathbf{j}\in\mathcal{K}(d-1,K).$$

Then, the boundary terms as appearing in (5.5), on each face, read

$$\begin{aligned}\Upsilon^{(\pm\alpha)} &:= -\dot{\mathbf{e}}^T\mathbf{B}_{\text{quad}}^{(\pm\alpha)}\dot{\mathbf{e}} + \dot{\mathbf{e}}^T\mathbf{g}^{(\pm\alpha),\text{quad}} + \dot{\mathbf{e}}^T\mathbf{g}_\mathcal{K}^{(\pm\alpha)} \\ &= -\dot{\mathbf{e}}^T\mathbf{B}_{\text{quad}}^{(\pm\alpha)}\dot{\mathbf{e}} - \dot{\mathbf{e}}^T\left(\mathbf{B}^{(\pm\alpha)} - \mathbf{B}_{\text{quad}}^{(\pm\alpha)}\right)\dot{\mathbf{u}} + \dot{\mathbf{e}}^T\left(\mathbf{H}^{(\pm\alpha)}\right)^T\widehat{\mathbf{g}}_\mathcal{K}^{(\pm\alpha)} \\ &= -\left(\mathbf{H}^{(\pm\alpha)}\dot{\mathbf{e}}\right)^T\widehat{\mathbf{B}}_{\text{quad}}^{(\pm\alpha)}\mathbf{H}^{(\pm\alpha)}\dot{\mathbf{e}} \\ &\quad - \left(\mathbf{H}^{(\pm\alpha)}\dot{\mathbf{e}}\right)^T\left(\widehat{\mathbf{B}}^{(\pm\alpha)} - \widehat{\mathbf{B}}_{\text{quad}}^{(\pm\alpha)}\right)\mathbf{H}^{(\pm\alpha)}\dot{\mathbf{u}} + \left(\mathbf{H}^{(\pm\alpha)}\dot{\mathbf{e}}\right)^T\widehat{\mathbf{g}}_\mathcal{K}^{(\pm\alpha)}.\end{aligned}$$

ETNA
Kent State University
http://etna.math.kent.edu

The positivity condition on $b^{(\pm\alpha)}(x)$ gives $\mathbf{v}^T\widehat{\mathbf{B}}^{(\pm\alpha)}_{\text{quad}}\mathbf{v} \geq b_0\mathbf{v}^T\mathbf{v}$. We are then able to bound the contribution from $\mathbf{g}^{(\pm\alpha),\text{quad}}$ by completing the square,

$$\begin{aligned}\Upsilon^{(\pm\alpha)} &\leq -b_0\left(\mathbf{H}^{(\pm\alpha)}\dot{\mathbf{e}}\right)^T\mathbf{H}^{(\pm\alpha)}\dot{\mathbf{e}} + \left(\mathbf{H}^{(\pm\alpha)}\dot{\mathbf{e}}\right)^T\left(-\left(\widehat{\mathbf{B}}^{(\pm\alpha)} - \widehat{\mathbf{B}}^{(\pm\alpha)}_{\text{quad}}\right)\mathbf{H}^{(\pm\alpha)}\dot{\mathbf{u}} + \widehat{\mathbf{g}}^{(\pm\alpha)}_{\mathcal{K}}\right)\\ &\leq \frac{1}{4b_0}\left\|-\left(\widehat{\mathbf{B}}^{(\pm\alpha)} - \widehat{\mathbf{B}}^{(\pm\alpha)}_{\text{quad}}\right)\mathbf{H}^{(\pm\alpha)}\dot{\mathbf{u}} + \widehat{\mathbf{g}}^{(\pm\alpha)}_{\mathcal{K}}\right\|^2.\end{aligned}$$

Using a $(d-1)$-dimensional analogue of the projection $\mathbf{P}$, we find

$$\widehat{\mathbf{B}}^{(\pm\alpha)} - \widehat{\mathbf{B}}^{(\pm\alpha)}_{\text{quad}} = \mathbf{P}^{(\neg\alpha)}\left(\widehat{\mathbf{B}}^{(\pm\alpha)} - \widehat{\mathbf{B}}^{(\pm\alpha)}_{\text{quad}}\right)\mathbf{P}^{(\neg\alpha)}.$$

Together with the bound on $b(x)$ and the projection estimate (5.4), this implies

$$\begin{aligned}\left\|\left(\widehat{\mathbf{B}}^{(\pm\alpha)} - \widehat{\mathbf{B}}^{(\pm\alpha)}_{\text{quad}}\right)\mathbf{H}^{(\pm\alpha)}\dot{\mathbf{u}}\right\| &\leq \left(\left\|\widehat{\mathbf{B}}^{(\pm\alpha)}\right\| + \left\|\widehat{\mathbf{B}}^{(\pm\alpha)}_{\text{quad}}\right\|\right)\left\|\mathbf{P}^{(\neg\alpha)}\mathbf{H}^{(\pm\alpha)}\dot{\mathbf{u}}\right\|\\ &\leq 2Cb_{\max}\left(\|\mathcal{P}^{(\neg\alpha)}u_t\|_{\Omega^{(\pm\alpha)}} + \|\mathcal{P}^{\perp}_{\mathcal{K}}u_t\|_{\Omega^{(\pm\alpha)}}\right).\end{aligned}$$

We add the contributions from the different boundary faces together and apply the trace inequality, to get

$$\begin{aligned}\left\|\left(\widehat{\mathbf{B}} - \widehat{\mathbf{B}}_{\text{quad}}\right)\sum_{\pm\alpha}\mathbf{H}^{(\pm\alpha)}\dot{\mathbf{u}}\right\| &\leq Cb_{\max}\left(\left\|\sum_{\pm\alpha}\mathcal{P}^{(\neg\alpha)}u_t\right\|_{\partial\Omega} + \left\|\mathcal{P}^{\perp}_{\mathcal{K}}u_t\right\|_{\partial\Omega}\right)\\ &\leq Cb_{\max}\left(\left\|\sum_{\pm\alpha}\mathcal{P}^{(\neg\alpha)}u_t\right\|_{H^1(\Omega)} + \left\|\mathcal{P}^{\perp}_{\mathcal{K}}u_t\right\|_{H^1(\Omega)}\right)\\ &\leq Cb_{\max}(K-R)^{5/2-s}|u_t|_{H^{s-1}(\Omega)}.\end{aligned}$$

We also have

$$\begin{aligned}\|\widehat{\mathbf{g}}_{\mathcal{K}}\| &\leq \|g_{\mathcal{K}}\|_{\partial\Omega} \leq \|\mathbf{n}\cdot a\nabla\mathcal{P}^{\perp}_{\mathcal{K}}u\|_{\partial\Omega} + \|b\mathcal{P}^{\perp}_{\mathcal{K}}u_t\|_{\partial\Omega}\\ &\leq da_{\max}\|\mathcal{P}^{\perp}_{\mathcal{K}}u\|_{H^1(\partial\Omega)} + b_{\max}\|\mathcal{P}^{\perp}_{\mathcal{K}}u_t\|_{\partial\Omega}\\ &\leq da_{\max}\|\mathcal{P}^{\perp}_{\mathcal{K}}u\|_{H^2(\Omega)} + b_{\max}\|\mathcal{P}^{\perp}_{\mathcal{K}}u_t\|_{H^1(\Omega)}\\ &\leq CK^{7/2-s}|u|_{H^s(\Omega)} + CK^{5/2-s}|u_t|_{H^{s-1}(\Omega)}.\end{aligned}$$

Thereby,

$$-\dot{\mathbf{e}}^T\mathbf{B}_{\text{quad}}\dot{\mathbf{e}} + \dot{\mathbf{e}}^T\left(\mathbf{g}^{\text{quad}} + \mathbf{g}_{\mathcal{K}}\right) \leq CK^{7-2s}|u|^2_{H^s(\Omega)} + C(K-R)^{5-2s}|u_t|^2_{H^{s-1}(\Omega)}.$$

Integrating (5.5) and putting everything together proves the lemma. ∎

6. Numerical experiments. We demonstrate our method using our usual example, the wave equation in d dimensions, where $d = 2, 3$. In Cartesian coordinates and with an isotropic medium, the problem reads

$$\begin{aligned}u_{tt} &= \Delta u, && y \in \tilde{\Omega} \subset \mathbb{R}^d, \quad t \geq 0,\\ u(y,0) &= u_t(y,0) = 0, && y \in \tilde{\Omega},\\ \tilde{b}u_t + \mathbf{n}\cdot\nabla u &= \tilde{g}(y,t), && y \in \partial\tilde{\Omega}.\end{aligned} \tag{6.1}$$

ETNA
Kent State University
http://etna.math.kent.edu

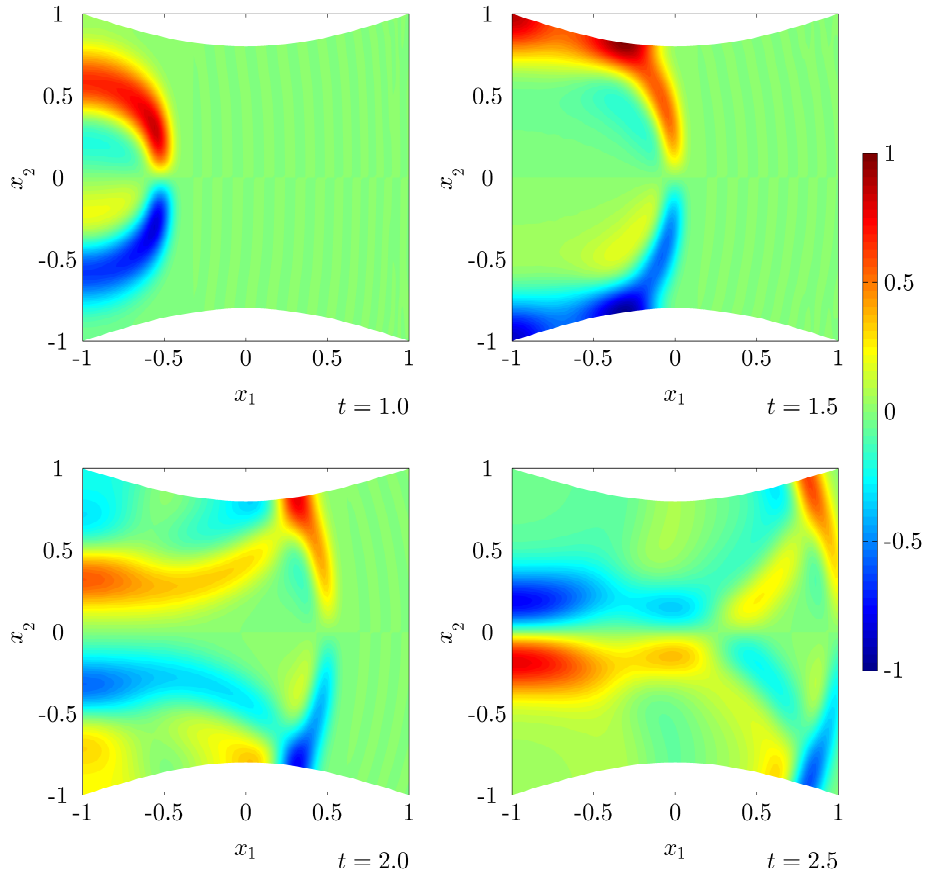

FIG. 6.1. *The time evolution of* (6.1) *for* $d = 2$.

We use homogeneous initial data, and excite the problem through the boundary condition. We assume that there exists a smooth coordinate transformation $x = x(y)$ which maps $\tilde{\Omega} \mapsto \Omega = [-1, 1]^d$, and that the corresponding Jacobian matrix is positive definite,

$$J_{j,k} = \frac{\partial y_j}{\partial x_k}, \qquad \gamma = \det(J) \geq \gamma_0 > 0 \quad \forall x \in \Omega.$$

We can then reformulate the problem on the hypercube as

$$\begin{aligned} \gamma u_{tt} &= \nabla \cdot a \nabla u, && x \in \Omega,\ t \geq 0, \\ u(x,0) &= u_t(x,0) = 0, && x \in \Omega, \\ b u_t + \mathbf{n} \cdot a \nabla u &= g(x,t), && x \in \partial\Omega, \end{aligned}$$

with $a = \gamma J^{-1} J^{-T}$ and $b = \delta \tilde{b}$, $g = \delta \tilde{g}$, where $\delta = \gamma \| J^{-T} \mathbf{n} \|$. When $d = 2$, we use Neumann boundary conditions, $\mathbf{n} \cdot a \nabla u = 0$, on $x_2 = \pm 1$, and the first order Engquist–Majda [11] absorbing boundary condition, $\delta u_t + \mathbf{n} \cdot a \nabla u = 0$, on $x_1 = 1$. On the $x_1 = -1$ boundary, we use the boundary condition $\mathbf{n} \cdot a \nabla u = g(x,t)$, with

$$\tilde{g}(x,t) = M x_2 \exp\left(-20 x_2^2 - 200 (t - t_0)^2\right), \qquad M = 200,\ t_0 = \frac{1}{2}.$$

ETNA
Kent State University
http://etna.math.kent.edu

We solve the problem on the domain

$$\tilde{\Omega} = \{(y_1, y_2) \ : \ -1 + \phi(y_1) \le y_2 \le 1 - \phi(y_1), \ -1 \le y_1 \le 1\},$$

with $\phi(y_1) = 0.2\cos(\pi y_1)$. For $d = 3$, we extrude the domain to $-1 \le y_3 \le 1$. We then use homogeneous Neumann boundary conditions on $x_3 = \pm 1$ as well as on $x_2 = \pm 1$. The first order Engquist–Majda boundary condition on $x_1 = 1$ has a direct generalisation to three dimensions, and the boundary forcing on $x_1 = -1$ is extended as

$$\tilde{g}(x,t) = M^2 x_2 x_3 \exp\left(-20x_2^2 - 20x_3^2 - 200(t - t_0)^2\right), \qquad M = 200, \ t_0 = \frac{1}{2}.$$

With the same positivity assumptions on a and b as previously, this problem satisfies an energy estimate of the form (4.1) in the γ-weighted energy norm

$$\|u\|_{E,\gamma} = \left((u_t, \gamma u_t) + (\nabla u, a \nabla u)\right)^{1/2},$$

cf. (4.2). However, the energy estimate blows up when $b_0 \to 0$. As we have chosen $b = 0$ on parts of the boundary, we cannot show strong well-posedness for this problem using energy estimates alone. Strong well-posedness can nevertheless be proven by combining an energy estimate, which exists for $b = 0$ and $g = 0$, with the Laplace–Fourier technique, as described in [17, Ch. 10].

After spatial discretisation, we get a system of ODEs of the form

$$\mathbf{\Gamma}\ddot{\mathbf{v}} = -\mathbf{S}\mathbf{v} - \mathbf{B}\dot{\mathbf{v}} + \mathbf{g}, \tag{6.2}$$

where $\mathbf{\Gamma} = \gamma_{\text{pol}}(\mathbf{X})$ denotes formal insertion of the coordinate matrices in a polynomial approximation of $\gamma(x)$. $\mathbf{\Gamma}^{-1}$ is computed as $(\gamma^{-1})_{\text{pol}}(\mathbf{X})$, rather than as the inverse of $\mathbf{\Gamma}$. We let $\mathbf{w} = \dot{\mathbf{v}}$ and rewrite (6.2) as the first order system

$$\begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\mathbf{w}} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -\mathbf{\Gamma}^{-1}\mathbf{S} & -\mathbf{\Gamma}^{-1}\mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{\Gamma}^{-1}\mathbf{g} \end{bmatrix}. \tag{6.3}$$

A first order formulation facilitates explicit time-stepping using standard methods, e.g., the 4th order Runge–Kutta method. All constituent parts of the right-hand side can be evaluated in essentially linear time using the procedures devised in this paper.

In Figure 6.1, we show what the solution of the two-dimensional problem looks like at four different points in time. Note how the solution is reflected by, as well as glancing along, the curved boundaries. The computation was done with order $K = 48$ Legendre polynomials in each direction, and the coefficients a, b and γ^{-1} were represented by their order $R = 6$ Chebyshev interpolants. Time-stepping was done with the 4th order Runge–Kutta method with the time step $\Delta t = 0.002$.

To test the spatial accuracy of our method, in view of Theorem 5.1, we propagate (6.3) using the 4th order Runge–Kutta method and measure the L^2-error of the obtained approximations as a function of K. The fixed choice of time step size $\Delta t = 0.001$ is sufficient to guarantee stability for all subsequent choices of K, and the error contribution from spatial discretisation is always dominant. The reference solution is computed with $K = 70$ and $\Delta t = 0.0005$. Figure 6.2 shows the observed errors and propagation times in two and three dimensions. As is readily seen, the observed errors decay exponentially with respect to K. In a practical computation, one would let the time step scale with the spectrum of the right-hand side matrix, following the stability condition of the time-stepping scheme. In the present example, $\|\mathbf{S}\| \sim K^4$, cf. (5.7), which results in the spectral norm of the matrix in (6.3) being

ETNA
Kent State University
http://etna.math.kent.edu

K	$d=2$		$d=3$	
	error	time (s)	error	time (h)
22	3.511e-02	25.3	3.754e-01	1.32
26	2.095e-02	35.4	2.347e-01	2.21
30	1.427e-02	47.1	1.527e-01	3.38
34	7.136e-03	60.4	7.696e-02	4.91
38	3.799e-03	75.5	4.387e-02	6.84
42	2.235e-03	92.3	2.430e-02	9.22
46	9.808e-04	110.6	1.066e-02	12.10
50	4.879e-04	131.1	5.621e-03	15.53
54	2.542e-04	152.6	2.746e-03	19.56
58	9.413e-05	176.0	1.042e-03	24.25
62	4.684e-05	201.0	5.305e-04	29.63

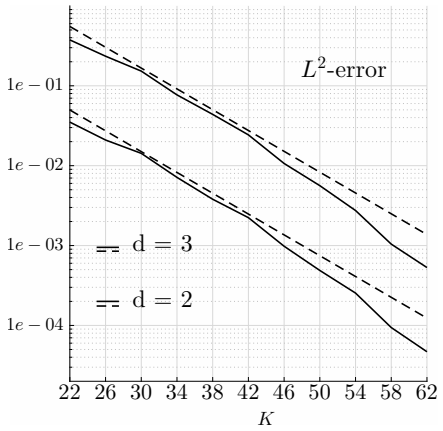

FIG. 6.2. *L^2-errors and computation times for both $d=2$ (black lines) and $d=3$ (grey lines) for various choices of K. The solid lines represent the L^2-errors, while the dashed lines stand for curves that are proportional to $\exp(-0.15K)$. Semilogarithmic plot. Using a fixed time step for all spatial resolutions, we verify that the computation times are approximately linear in K^d.*

of size $\sim K^2$. The time step restriction for an explicit method is therefore quadratic in $1/K$. The computations were done on a desktop computer with an Intel Xeon E5-1620 processor at 3.5 GHz and with 8 GB memory.

As a conclusion, we briefly comment on how to explicitly assemble the stiffness matrix $\mathbf{S}_{\text{quad}}$ using the $(K+1)$-node Gauß–Legendre quadrature rule in each dimension, and compare this to the above fast algorithm. The entries of $\mathbf{S}_{\text{quad}}$ are best computed by the quadrature rule

$$(\mathbf{S}_{\text{quad}})_{\mathbf{j},\mathbf{k}} = \sum_{\alpha,\beta=1}^{d} \sum_{\mathbf{r}\in\mathcal{R}} \alpha_{\mathbf{r}}^{(\alpha,\beta)} \prod_{\gamma=1}^{d} \left[\sum_{m_\gamma=0}^{K} \omega_{m_\gamma} \varphi_{j_\gamma}^{(\alpha,\gamma)}(\xi_{m_\gamma}) T_{r_\gamma}(\xi_{m_\gamma}) \varphi_{k_\gamma}^{(\beta,\gamma)}(\xi_{m_\gamma}) \right], \tag{6.4}$$

with $\varphi_j^{(\alpha,\gamma)} = \frac{\partial}{\partial x_\alpha}\varphi_j$ in case $\alpha=\gamma$, and $\varphi_j^{(\alpha,\gamma)} = \varphi_j$ otherwise. First, one computes $\varphi_k'(\xi_j)$ and $T_r(\xi_j)$ at the Gaussian quadrature nodes, for all $0 \le j,k \le K$ and $0 \le r \le R$. Using the recurrence relation for the derivatives of the Legendre basis functions,

$$\varphi_k'(x) = \sqrt{(2k+1)(2k-1)}\varphi_{k-1}(x) + \sqrt{\frac{2k+1}{2k-3}}\varphi_{k-2}'(x), \qquad k \ge 2,$$
$$\varphi_0'(x) = 0, \qquad \varphi_1'(x) = \sqrt{\frac{3}{2}},$$

in combination with (2.5), this requires $\mathcal{O}(K^2)$ and $\mathcal{O}(RK)$ operations, respectively. Second, from these values, the sum in square brackets as occurring in (6.4) is computed for all choices of j_γ, k_γ, and r_γ in $\mathcal{O}(K^3R)$. Finally, given these quantities and interpolation coefficients $\alpha_{\mathbf{r}}^{(\alpha,\beta)}$, $\mathbf{r} \in \mathcal{R}(d,R)$, for $a_{\text{pol}}^{(\alpha,\beta)}$, $\alpha,\beta = 1,\dots,d$, the assembly of $\mathbf{S}_{\text{quad}}$ according to (6.4) requires $\sim d^3|\mathcal{R}||\mathcal{K}|^2 \sim d^3R^dK^{2d}$ operations. When all matrices are assembled, the cost of time-stepping over a fixed time-interval is $\mathcal{O}(K^{2d+2})$ operations. This contrasts to the total computational costs for solving the problem over a fixed time-interval using our matrix-free method, which scale as $\sim d^2R^dK^{d+2}$, cf. Section 2.5.

7. Conclusion. We have presented a Legendre spectral method for initial-boundary value problems with general boundary conditions and variable coefficients in arbitrary dimensions,

ETNA
Kent State University
http://etna.math.kent.edu

where the computational work in each time step scales only linearly with the number of unknowns. The proposed methodology is particularly relevant for problems with boundary conditions such that spectral bases that allow for fast transforms between function values and expansion coefficients are inappropriate. We combine previously known recursive procedures for evaluating derivatives with newly devised procedures for variable coefficients. This contrasts to an explicit assembly of the (dense) matrices due to spectral discretisation, the latter being prohibitively expensive. The effect of variable coefficients is applied using appropriately modified ideas from [4, 12]. Boundary conditions are enforced weakly. We have analysed the error, which is seen to decay spectrally. In the paper we have used the scalar wave equation as model problem, both for analysis and numerical experiments.

The method has been tested numerically in two and three dimensions, in curvilinear coordinates. In the case of the scalar wave equation, a time step restriction $\Delta t \sim K^{-2}$ is required for stability of standard explicit schemes. This is comparably restrictive. For finite difference and finite element methods, the time step restriction is linear in the spatial step size. One possibility for mitigating this disadvantage is exponential time integration [21]. The approximation of matrix exponentials and related matrix functions with Krylov subspace method is based on matrix-vector multiplication, where the present method can be applied directly.

Adaptation of the method to several other classes of linear initial-boundary value problems is immediate, analogously to discontinuous Galerkin and SBP finite difference methods. For parabolic problems, where explicit time-stepping is inappropriate, we would advise using exponential time integration. The method is designed for time-dependent problems. For elliptic problems one could solve the corresponding parabolic problem to steady-state, or use one of the related techniques [9, 27, 35].

Acknowledgements. Most of the research reported in this paper was carried out while E.K. was visiting Universität Tübingen with support from Anna Maria Lundins stipendiefond vid Smålands nation i Uppsala. B.B. is funded by the DFG Priority Program 1324 and is associated with DFG Research Training Group 1838. The authors are grateful to Christian Lubich for valuable suggestions and helpful discussions, and thank the anonymous referees for comments which significantly improved the manuscript.

REFERENCES

[1] I. BABUŠKA AND M. SURI, *The p and h-p versions of the finite element method, basic principles and properties*, SIAM Rev., 36 (1994), pp. 578–632.
[2] S. BEUCHLER AND J. SCHÖBERL, *New shape functions for triangular p-FEM using integrated Jacobi polynomials*, Numer. Math., 103 (2006), pp. 339–366.
[3] J. P. BOYD, *Chebyshev and Fourier Spectral Methods*, 2nd ed., Dover, Mineola, 2001.
[4] B. BRUMM, *A fast matrix-free algorithm for spectral approximations to the Schrödinger equation*, SIAM J. Sci. Comput., 37 (2015), pp. A2003–A2025.
[5] C. CANUTO, M. Y. HUSSAINI, A. QUARTERONI, AND T. A. ZANG, *Spectral Methods: Fundamentals in Single Domains*, Springer, Berlin, 2006.
[6] M. H. CARPENTER AND D. GOTTLIEB, *Spectral methods on arbitrary grids*, J. Comput. Phys., 129 (1996), pp. 74–86.
[7] M. H. CARPENTER, D. GOTTLIEB, AND S. ABARBANEL, *Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: methodology and application to high-order compact schemes*, J. Comput. Phys., 111 (1994), pp. 220–236.
[8] C. W. CLENSHAW, *The numerical solution of linear differential equations in Chebyshev series*, Proc. Cambridge Philos. Soc., 53 (1957), pp. 134–149.
[9] E. A. COUTSIAS, T. HAGSTROM, AND D. TORRES, *An efficient spectral method for ordinary differential equations with rational function coefficients*, Math. Comp., 65 (1996), pp. 611–635.
[10] D. A. DI PIETRO AND A. ERN, *Mathematical Aspects of Discontinuous Galerkin Methods*, Springer, Heidelberg, 2012.

ETNA
Kent State University
http://etna.math.kent.edu

[11] B. ENGQUIST AND A. MAJDA, *Absorbing boundary conditions for the numerical simulation of waves*, Math. Comp., 31 (1977), pp. 629–651.
[12] E. FAOU, V. GRADINARU, AND C. LUBICH, *Computing semiclassical quantum dynamics with Hagedorn wavepackets*, SIAM J. Sci. Comput., 31 (2009), pp. 3027–3041.
[13] K.-A. FENG, C.-H. TENG, AND M.-H. CHEN, *A pseudospectral penalty scheme for 2D isotropic elastic wave computations*, J. Sci. Comput., 33 (2007), pp. 313–348.
[14] D. FUNARO, *Domain decomposition methods for pseudo spectral approximations. Part I. Second order equations in one dimension*, Numer. Math., 52 (1988), pp. 329–344.
[15] D. FUNARO AND D. GOTTLIEB, *A new method of imposing boundary conditions in pseudospectral approximations of hyperbolic equations*, Math. Comp., 51 (1988), pp. 599–613.
[16] M. J. GROTE, A. SCHNEEBELI, AND D. SCHÖTZAU, *Discontinuous Galerkin finite element method for the wave equation*, SIAM J. Numer. Anal., 44 (2006), pp. 2408–2431.
[17] B. GUSTAFSSON, H.-O. KREISS, AND J. OLIGER, *Time-Dependent Problems and Difference Methods*, 2nd ed., Wiley, Hoboken, 2013.
[18] J. S. HESTHAVEN, *Spectral penalty methods*, Appl. Numer. Math. 33 (2000), pp. 23–41.
[19] J. S. HESTHAVEN AND D. GOTTLIEB, *A stable penalty method for the compressible Navier-Stokes equations. I. Open boundary conditions*, SIAM J. Sci. Comput., 17 (1996), pp. 579–612.
[20] J. S. HESTHAVEN AND T. WARBURTON, *Nodal Discontinuous Galerkin Methods. Algorithms, Analysis, and Applications*, Springer, New York, 2008.
[21] M. HOCHBRUCK AND A. OSTERMANN, *Exponential integrators*, Acta Numer., 19 (2010), pp. 209–286.
[22] C. KOUTSCHAN, C. LEHRENFELD, AND J. SCHÖBERL, *Computer algebra meets finite elements: An efficient implementation for Maxwell's equations*, in Numerical and Symbolic Scientific Computing, U. Langer and P. Paule, eds., Texts & Monographs in Symbolic Computation, Springer, Wien, 2012, pp. 105–121.
[23] H.-O. KREISS AND J. OLIGER, *Stability of the Fourier method*, SIAM J. Numer. Anal., 16 (1979), pp. 421–433.
[24] K. MATTSSON, F. HAM, AND G. IACCARINO, *Stable boundary treatment for the wave equation on second-order form*, J. Sci. Comput., 41 (2009), pp. 366–383.
[25] J. M. MELENK, K. GERDES, AND C. SCHWAB, *Fully discrete hp-finite elements: fast quadrature*, Comput. Methods Appl. Mech. Engrg., 190 (2001), pp. 4339–4364.
[26] J. T. ODEN, I. BABUŠKA, AND C. E. BAUMANN, *A discontinuous hp finite element method for diffusion problems*, J. Comput. Phys., 146 (1998), pp. 491–519.
[27] S. OLVER AND A. TOWNSEND, *A fast and well-conditioned spectral method*, SIAM Rev., 55 (2013), pp. 462–489.
[28] S. A. ORSZAG, *Spectral methods for problems in complex geometries*, J. Comput. Phys., 37 (1980), pp. 70–92.
[29] W. H. PRESS, S. A. TEUKOLSKY, W. T. VETTERLING, AND B. P. FLANNERY, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed., Cambridge University Press, Cambridge, 1992.
[30] G. SCHERER, *On Energy Estimates for Difference Approximations to Hyperbolic Partial Differential Equations*, PhD Thesis, Department of Computer Sciences, Uppsala University, 1977.
[31] C. SCHWAB, *p- and hp- Finite Element Methods. Theory and Applications in Solid and Fluid Mechanics*, Oxford University Press, New York, 1998.
[32] J. SHEN, T. TANG, AND L.-L. LIAN, *Spectral Methods: Algorithms, Analysis and Applications*, Springer, Heidelberg, 2011.
[33] E. TADMOR, *The exponential accuracy of Fourier and Chebyshev differencing methods*, SIAM J. Numer. Anal., 23 (1986), pp. 1–10.
[34] V. THOMÉE, *Galerkin Finite Element Methods for Parabolic Problems*, 2nd ed., Springer, Berlin, 1997.
[35] A. TOWNSEND AND S. OLVER, *The automatic solution of partial differential equations using a global spectral method*, J. Comput. Phys., 299 (2015), pp. 106–123.