



An Adaptive Multiresolution Interior Penalty Discontinuous Galerkin Method for Wave Equations in Second Order Form

Juntao Huang¹ · Yuan Liu² · Wei Guo³ · Zhanjing Tao⁴ · Yingda Cheng⁵

Received: 17 April 2020 / Revised: 21 August 2020 / Accepted: 21 September 2020 /

Published online: 4 October 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

In this paper, we propose a class of adaptive multiresolution (also called adaptive sparse grid) discontinuous Galerkin (DG) methods for simulating scalar wave equations in second order form in space. The two key ingredients of the schemes include an interior penalty DG formulation in the adaptive function space and two classes of multiwavelets for achieving multiresolution. In particular, the orthonormal Alpert's multiwavelets are used to express the DG solution in terms of a hierarchical structure, and the interpolatory multiwavelets are further introduced to enhance computational efficiency in the presence of variable wave speed or nonlinear source. Some theoretical results on stability and accuracy of the proposed method are presented. Benchmark numerical tests in 2D and 3D are provided to validate the performance of the method.

Keywords Sparse grid · Multiresolution · Interior penalty discontinuous Galerkin method · Wave equation · Adaptivity

1 Introduction

Wave propagation, governed by the wave equation, is ubiquitous in science and engineering, such as sound waves, light waves, and water waves propagating in acoustics, electromagnetics and geoscience. Designing efficient and robust numerical methods to solve the wave equation is of fundamental and practical importance in those applications. The goal of this work is to design a class of numerical solvers that are adaptive, high order accurate, and more importantly, work efficiently in high dimensions. In particular, we develop a class of adaptive multiresolution (also called adaptive sparse grid) discontinuous Galerkin (DG) method for

Research supported in part by a grant from the Simons Foundation (426993, Yuan Liu). Research is supported by NSF Grant DMS-1830838, Research is supported by NSF Grants DMS-1453661 and DMS-1720023. Research is supported by NSFC Grant 12001231 (Zhanjing Tao).

✉ Zhanjing Tao
zjtao@jlu.edu.cn

Extended author information available on the last page of the article

the following model second-order wave equation

$$u_{tt} = \nabla \cdot (c^2(\mathbf{x}) \nabla u) + f \quad (1.1)$$

on the bounded domain $\Omega = [0, 1]^d$ in arbitrary d dimensions, subject to initial conditions

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad u_t(\mathbf{x}, 0) = v_0(\mathbf{x}). \quad (1.2)$$

We assume that the wave speed $c(\mathbf{x})$ is piecewise smooth and bounded below and above uniformly, i.e., $0 < C_* \leq c^2(\mathbf{x}) \leq C^* < \infty$. For simplicity, we only consider periodic, Dirichlet and Neumann boundary conditions in this paper. Extensions to more complicated domains and other types of boundary conditions will be considered in the future work.

A vast amount of numerical methods have been developed in the literature on the numerical approximations of the wave equation, including finite difference discretization [16, 23, 24, 36], spectral and spectral element discretization [18, 33, 38] and finite element discretization [1, 29], to name a few. As a special class of finite element discretization, the DG methods [15, 32] have become very popular recently in approximating partial differential equations (PDEs) due to their distinguished advantages in handling geometry, boundary conditions and accommodating adaptivity. In the context of the wave simulations, DG methods have been successfully developed for simulating wave equations in first-order form [25, 30, 40], second-order form [3, 13, 20, 41], and with hp -adaptivity [17]. In this paper, we utilize the symmetric interior penalty DG (IPDG) method [4] for wave equation in second order form [20], though our framework can work with other types of DG schemes.

Adaptivity is crucial for efficient simulations of the wave equation due to the multiscale nature of the solution structures. The well-known adaptive mesh refinement (AMR) [6, 7] adjusts the computational grid adaptively to track small scale features of the underlying problems, improving computational efficiency significantly. AMR has been incorporated in various software framework and packages to simulate wave propagation with great success [9, 11]. In contrast, this paper considers adaptive simulations in the multiresolution sense. The main idea of multiresolution analysis (MRA) [31] is to explore mesh hierarchy, which induces nested polynomial approximation spaces to accelerate the computation and in the mean time circumvents the need for *a posteriori* error indicators. MRA is also the foundation of sparse grid methods [10], which is known as a popular dimension reduction technique for solving high dimensional problems. As a continuation of our previous research for adaptive multiresolution (also called adaptive sparse grid) DG methods [22, 27] for first order equations, this paper develops an adaptive multiresolution IPDG solver for 2D and 3D scalar wave equations (1.1). In particular, we employ the Alpert's multiwavelets as the DG bases in the IPDG formulation, following the approach proposed in [21, 22, 39] for linear equations, together with the interpolatory multiwavelets for efficiently computing variable wave speed problems as done in [27] for nonlinear hyperbolic conservation laws. We refer the readers to [27] for more details on the background of adaptive multiresolution DG methods [12, 26]. It is worth noting that a fast matrix-vector multiplication algorithm [34, 42] is essential for efficient implementation of the method with varying wave speed. We conduct error analysis for the semi-discrete formulation for the scheme with and without interpolations. First, when the sparse grid piecewise polynomial space of degree k is employed in the IPDG formulation as in [39], the newly proposed method converges with order k and a polylogarithmic factor in the energy norm for sufficiently smooth problems with constant coefficients. Second, in the case of smooth problems with variable coefficients, the proposed interpolatory technique ensures a high order local truncation error and hence preserve the original accuracy of the scheme given sufficient high order accuracy of interpolation. Numerical experiments in 2D

and 3D verify the accuracy of the methods. In particular, the adaptive scheme is demonstrated to capture the fine scale structure presented in inhomogeneous media.

The rest of the paper is organized as follows. In Sect. 2, we review Alpert's and interpolatory multiwavelets. Section 3 describes the numerical schemes with details on some theoretical results and implementations. Section 4 contains numerical examples. In Sect. 5, we make conclusions and discuss future work. Appendix collects detailed formulas of interpolatory multiwavelets used in this paper.

2 MRA and Multiwavelets

In this section, we first review the L^2 orthonormal Alpert's multiwavelets [2] and the sparse grid DG finite element space [21,39]. Next, we review the interpolatory multiwavelets proposed in [37], which has been used for the calculation of nonlinear conservation laws in [27].

2.1 Alpert's Multiwavelets

In this subsection, we review the construction of sparse grid DG finite element space based on Alpert's multiwavelets [2]. For a unit domain $\Omega = [0, 1]$ in 1D, we define a set of nested grids, where the n -th level grid Ω_n consists of 2^n uniform cells

$$I_n^j = (2^{-n}j, 2^{-n}(j+1)], \quad j = 0, \dots, 2^n - 1$$

for $n \geq 0$. For notational convenience, we also denote $I_{-1} = [0, 1]$. The piecewise polynomial space of degree at most $k \geq 1$ on the n -th level grid Ω_n for $n \geq 0$ is denoted by

$$V_n^k := \{v : v \in P^k(I_n^j), \forall j = 0, \dots, 2^n - 1\}. \quad (2.1)$$

Because of the nested structure

$$V_0^k \subset V_1^k \subset V_2^k \subset V_3^k \subset \dots,$$

we define the multiwavelet subspace W_n^k , $n = 1, 2, \dots$ as the orthogonal complement of V_{n-1}^k in V_n^k with respect to the L^2 inner product on $[0, 1]$, i.e.,

$$V_{n-1}^k \oplus W_n^k = V_n^k, \quad W_n^k \perp V_{n-1}^k.$$

Denote $W_0^k := V_0^k$, we have $V_n^k = \bigoplus_{0 \leq l \leq n} W_l^k$. A set of orthonormal basis can be defined on W_l^k as follows. When $l = 0$, the basis $v_{i,0}^0(x)$, $i = 0, \dots, k$ are the normalized shifted Legendre polynomials in $[0, 1]$. When $l > 0$, the Alpert's orthonormal multiwavelets are employed [2] as the bases and denoted by

$$v_{i,l}^j(x), \quad i = 0, \dots, k, \quad j = 0, \dots, 2^{l-1} - 1.$$

We then follow a tensor-product approach to construct the hierarchical finite element space in multi-dimensional space. Denote $\mathbf{l} = (l_1, \dots, l_d) \in \mathbb{N}_0^d$ as the mesh level in a multivariate sense, where \mathbb{N}_0 denotes the set of nonnegative integers, we can define the tensor-product mesh grid $\Omega_{\mathbf{l}} = \Omega_{l_1} \otimes \dots \otimes \Omega_{l_d}$ and the corresponding mesh size $h_{\mathbf{l}} = (h_{l_1}, \dots, h_{l_d})$.

Based on the grid $\Omega_{\mathbf{l}}$, we denote $I_{\mathbf{l}}^{\mathbf{j}} = \{\mathbf{x} : x_m \in (h_m j_m, h_m(j_m + 1)), m = 1, \dots, d\}$ as an elementary cell, and

$$\mathbf{V}_{\mathbf{l}}^k := \{\mathbf{v} : \mathbf{v} \in Q^k(I_{\mathbf{l}}^{\mathbf{j}}), \mathbf{0} \leq \mathbf{j} \leq 2^{\mathbf{l}} - \mathbf{1}\} = V_{l_1, x_1}^k \times \dots \times V_{l_d, x_d}^k$$

as the tensor-product piecewise polynomial space, where $Q^k(I_{\mathbf{l}}^{\mathbf{j}})$ represents the collection of polynomials of degree up to k in each dimension on cell $I_{\mathbf{l}}^{\mathbf{j}}$. If we use equal mesh refinement of size $h_N = 2^{-N}$ in each coordinate direction, the grid and space will be denoted by Ω_N and \mathbf{V}_N^k , respectively. Based on a tensor-product construction, the multi-dimensional increment space can be defined as

$$\mathbf{W}_{\mathbf{l}}^k = W_{l_1, x_1}^k \times \dots \times W_{l_d, x_d}^k.$$

The basis functions in multi-dimensions are defined as

$$v_{\mathbf{i}, \mathbf{l}}^{\mathbf{j}}(\mathbf{x}) := \prod_{m=1}^d v_{i_m, l_m}^{j_m}(x_m), \quad (2.2)$$

for $\mathbf{l} \in \mathbb{N}_0^d$, $\mathbf{j} \in B_{\mathbf{l}} := \{\mathbf{j} \in \mathbb{N}_0^d : \mathbf{0} \leq \mathbf{j} \leq \max(2^{\mathbf{l}-1} - \mathbf{1}, \mathbf{0})\}$ and $\mathbf{1} \leq \mathbf{i} \leq \mathbf{k} + \mathbf{1}$. The orthonormality of the bases can be easily verified.

Using the notation of

$$|\mathbf{l}|_1 := \sum_{m=1}^d l_m, \quad |\mathbf{l}|_\infty := \max_{1 \leq m \leq d} l_m.$$

and the same component-wise arithmetic operations and relations as defined in [39], we reach the decomposition

$$\mathbf{V}_N^k = \bigoplus_{\substack{|\mathbf{l}|_\infty \leq N \\ \mathbf{l} \in \mathbb{N}_0^d}} \mathbf{W}_{\mathbf{l}}^k. \quad (2.3)$$

On the other hand, a standard choice of sparse grid space [21,39] is

$$\hat{\mathbf{V}}_N^k = \bigoplus_{\substack{|\mathbf{l}|_1 \leq N \\ \mathbf{l} \in \mathbb{N}_0^d}} \mathbf{W}_{\mathbf{l}}^k \subset \mathbf{V}_N^k. \quad (2.4)$$

We skip the discussions on the details with regard to the property of the space, but refer the readers to [21,39]. In Sect. 3, we will describe the adaptive scheme which adapts a subspace of \mathbf{V}_N^k according to the numerical solution, hence offering more flexibility and efficiency.

2.2 Interpolatory Multiwavelets

Alpert's multiwavelets described in Sect. 2.1 are associated with the L^2 projection operator. The idea of interpolatory multiwavelet bases [37] is based on interpolation operators and is essential for the computation of variable coefficient problems. In this work, only Lagrange interpolation is considered, while we note that Hermite interpolation can be used. The details are provided below.

We define the set of interpolation points on the interval $I = [0, 1]$ at mesh level 0 by $X_0 = \{x_i\}_{i=0}^M \subset I$. Here, the number of points in X_0 is $(M + 1)$. We defer the discussion of the relations between M and k to Sect. 3.2.

The interpolation points at mesh level $n \geq 1$, X_n can be obtained correspondingly as

$$X_n = \{x_{i,n}^j := 2^{-n}(x_i + j), \quad i = 0, \dots, M, \quad j = 0, \dots, 2^n - 1\}.$$

We require the points to be nested, i.e.

$$X_0 \subset X_1 \subset X_2 \subset X_3 \subset \dots \quad (2.5)$$

This can be achieved by requiring $X_0 \subset X_1$.

Given the nodes, we define the basis functions on the zeroth level grid as Lagrange interpolation polynomials of degree $\leq M$ which satisfy the property:

$$\phi_i(x_{i'}) = \delta_{ii'},$$

for $i, i' = 0, \dots, M$. It is easy to see that $\text{span}\{\phi_i, i = 0, \dots, M\} = V_0^M$. With the basis function at mesh level zero, we can define the basis functions at mesh level $n \geq 1$:

$$\phi_{i,n}^j := \phi_i(2^n x - j), \quad i = 0, \dots, M, \quad j = 0, \dots, 2^n - 1$$

which form a complete basis set for V_n^M .

We now introduce the hierarchical representations and the interpolatory multiwavelets. Define $\tilde{X}_0 := X_0$ and $\tilde{X}_n := X_n \setminus X_{n-1}$ for $n \geq 1$, then we have the decomposition

$$X_n = \tilde{X}_0 \cup \tilde{X}_1 \cup \dots \cup \tilde{X}_n.$$

Denote the points in \tilde{X}_1 by $\tilde{X}_1 = \{\tilde{x}_i\}_{i=0}^M$. Then the points in \tilde{X}_n for $n \geq 1$ can be represented by

$$\tilde{X}_n = \{\tilde{x}_{i,n}^j := 2^{-(n-1)}(\tilde{x}_i + j), \quad i = 0, \dots, M, \quad j = 0, \dots, 2^{n-1} - 1\}.$$

For notational convenience, we let $\tilde{W}_0^M := V_0^M$. The increment function space \tilde{W}_n^M for $n \geq 1$ is introduced as a function space that satisfies

$$V_n^M = V_{n-1}^M \oplus \tilde{W}_n^M, \quad (2.6)$$

and is defined through the multiwavelets $\psi_i \in V_1^M$ that satisfies

$$\psi_i(x_{i'}) = 0, \quad \psi_i(\tilde{x}_{i'}) = \delta_{i,i'},$$

for $i, i' = 0, \dots, M$. Then \tilde{W}_n^M is given by

$$\tilde{W}_n^M = \text{span}\{\psi_{i,n}^j, \quad i = 0, \dots, M, \quad j = 0, \dots, 2^{n-1} - 1\}$$

where $\psi_{i,n}^j(x) := \psi_i(2^{n-1}x - j)$.

The multi-dimensional construction follows similar lines as in Sect. 2.1. We let

$$\tilde{\mathbf{W}}_1^M = \tilde{W}_{l_1,x_1}^M \times \dots \times \tilde{W}_{l_d,x_d}^M,$$

then

$$\mathbf{V}_N^M = \bigoplus_{\substack{|\mathbf{l}|_\infty \leq N \\ \mathbf{l} \in \mathbb{N}_0^d}} \tilde{\mathbf{W}}_{\mathbf{l}}^M,$$

while the sparse grid approximation space is

$$\hat{\mathbf{V}}_N^M = \bigoplus_{\substack{|\mathbf{l}|_1 \leq N \\ \mathbf{l} \in \mathbb{N}_0^d}} \tilde{\mathbf{W}}_{\mathbf{l}}^M.$$

Note that the construction by Alpert's multiwavelet and the interpolatory multiwavelet gives the same sparse grid space because of the same nested structure. Finally, the interpolation operator in multidimension is defined as $\mathcal{I}_N^M : C(\Omega) \rightarrow \mathbf{V}_N^M$:

$$\mathcal{I}_N^M[f](\mathbf{x}) = \sum_{\substack{|\mathbf{n}|_\infty \leq N \\ 0 \leq \mathbf{j} \leq \max(2^{\mathbf{n}-1} - 1, 0) \\ 0 \leq \mathbf{i} \leq \mathbf{M}}} b_{\mathbf{i}, \mathbf{n}}^{\mathbf{j}} \psi_{\mathbf{i}, \mathbf{n}}^{\mathbf{j}}(\mathbf{x}),$$

where the multi-dimensional basis functions $\psi_{\mathbf{i}, \mathbf{n}}^{\mathbf{j}}(\mathbf{x})$ are defined in the same approach as (2.2) by tensor products:

$$\psi_{\mathbf{i}, \mathbf{n}}^{\mathbf{j}}(\mathbf{x}) := \prod_{m=1}^d \psi_{i_m, n_m}^{j_m}(x_m). \quad (2.7)$$

For the sparse grid space $\hat{\mathbf{V}}_N^M$ or any adaptively chosen subspace of \mathbf{V}_N^M , the interpolation operator, which is denoted by \mathcal{I}_h in later sections, can be defined accordingly, by taking only multiwavelet basis functions that belong to that space. For completeness, we collect the detailed formulas of the interpolation points and the associated interpolatory multiwavelets used in this work in the Appendix.

3 Adaptive Multiresolution DG Scheme

In this section, we construct our numerical schemes for d -dimensional wave equation (1.1). We start by reviewing the semi-discrete IPDG formulation and its properties in Sect. 3.1. For variable wave speed, schemes with multiresolution interpolation are described in Sect. 3.2. Time stepping, adaptivity and fast implementations are discussed in Sects. 3.3 and 3.4.

3.1 Semi-Discrete Scheme

We use the IPDG formulation [20] for solving (1.1). Namely, we look for $u_h \in \mathbf{V}$, such that for any test function $v \in \mathbf{V}$,

$$\int_{\Omega} (u_h)_{tt} v \, d\mathbf{x} + B(u_h, v) = L(v). \quad (3.1)$$

where the bilinear form is defined as

$$\begin{aligned} B(u_h, v) &= \int_{\Omega} c^2 \nabla u_h \cdot \nabla v \, d\mathbf{x} - \sum_{e \in \Gamma} \int_e \{c^2 \nabla u_h\} \cdot [v] \, ds - \sum_{e \in \Gamma} \int_e \{c^2 \nabla v\} \cdot [u_h] \, ds \\ &\quad + \sum_{e \in \Gamma} \frac{\sigma}{h_N} \int_e [u_h] \cdot [v] \, ds \end{aligned} \quad (3.2)$$

and

$$L(v) = \int_{\Omega} f v \, d\mathbf{x} \quad (3.3)$$

for periodic or homogeneous Dirichlet boundary condition, and

$$L(v) = \int_{\Omega} f v \, d\mathbf{x} + \sum_{e \in \Gamma_D} \int_e (-c^2 \nabla v \cdot \mathbf{n} + \frac{\sigma}{h_N} v) g_D \, ds + \sum_{e \in \Gamma_N} \int_e g_N v \, ds \quad (3.4)$$

for Dirichlet and Neumann boundary conditions $u(\mathbf{x}, t)|_{\mathbf{x} \in \Gamma_D} = g_D$ and $\nabla u(\mathbf{x}, t) \cdot \mathbf{n}|_{\mathbf{x} \in \Gamma_N} = g_N$. Γ is the union of the boundaries for all the elements in the partition Ω_N , and σ is the penalty parameter depending on the dimension d . The average and jump are defined as,

$$\begin{aligned} [q] &= q^- \mathbf{n}^- + q^+ \mathbf{n}^+, \quad \{q\} = \frac{1}{2}(q^- + q^+), \\ [\mathbf{q}] &= \mathbf{q}^- \cdot \mathbf{n}^- + \mathbf{q}^+ \cdot \mathbf{n}^+, \quad \{\mathbf{q}\} = \frac{1}{2}(\mathbf{q}^- + \mathbf{q}^+). \end{aligned} \quad (3.5)$$

where \mathbf{n} is the unit normal. ‘-’ and ‘+’ represent that the directions of the vector point to interior and exterior at e respectively. If e is part of the boundary, then we let $[q] = q \mathbf{n}$ (\mathbf{n} is the outward unit normal) and $\{\mathbf{q}\} = \mathbf{q}$.

Depending on the choice of space \mathbf{V} , various IPDG methods with distinct properties are obtained. If $\mathbf{V} = \mathbf{V}_N^k$, we recover the IPDG scheme in [20] on tensor-product meshes. If $\mathbf{V} = \hat{\mathbf{V}}_N^k$, then we obtain the sparse grid IPDG method. If \mathbf{V} is chosen adaptively as described in Sect. 3.3, we have the adaptive multiresolution scheme. Note that besides the IPDG formulation, other DG formulations can be used as well, such as the local DG method [13] and the energy-based DG method [3]. The main novelty of this work is the choice of the multiresolution polynomial space which is not tied specifically to the weak formulation in use.

For completeness, we now review some properties of the semi-discrete IPDG scheme (3.1). Define the discrete energy of wave propagation by

$$E_h(t) := \frac{1}{2} \left\| \frac{\partial u_h}{\partial t} \right\|^2 + \frac{1}{2} B(u_h, u_h), \quad (3.6)$$

Then the stability inherently holds true since the bilinear form $B(\cdot, \cdot)$ is symmetric and coercive:

Theorem 3.1 (Energy stability [20]) *The discrete energy (3.6) is conserved by semi-discrete DG scheme (3.1)–(3.3) when $f = 0$ with periodic boundary condition for arbitrary choice of space including $\mathbf{V} = \mathbf{V}_N^k$ and $\mathbf{V} = \hat{\mathbf{V}}_N^k$.*

We then review some results in the error estimates [20], and extend it to the sparse grid method with $\mathbf{V} = \hat{\mathbf{V}}_N^k$ based on the approximation properties of the space $\hat{\mathbf{V}}_N^k$ in [21]. We use $\|\cdot\|$ to represent the standard L^2 norm on Ω or Ω_N , $\|\cdot\|_{L^2(\Gamma)}$ to represent the L^2 norm on the collection of the cell interfaces of the mesh $\Omega_N : \Gamma$, and define the energy norm of a function $v \in H^2(\Omega_N)$ as

$$\|v\|^2 := \int_{\Omega} |\nabla v|^2 d\mathbf{x} + \sum_{e \in \Gamma} h_N \int_e \left\{ \frac{\partial v}{\partial \mathbf{n}} \right\}^2 ds + \sum_{e \in \Gamma} \frac{1}{h_N} \int_e [v]^2 ds. \quad (3.7)$$

Some basic properties of the bilinear operator $B(\cdot, \cdot)$ are listed below.

Lemma 3.2 (Boundedness [4,5]) *There exists a positive constant C_b , depending only on C^* , σ , such that*

$$|B(w, v)| \leq C_b \|w\| \cdot \|v\|, \quad \forall w, v \in H^2(\Omega_N).$$

Lemma 3.3 (Coercivity [4,5]) *When σ is taken large enough, there exists a positive constant C_s depending only on C_* , such that*

$$B(v, v) \geq C_s \|v\|^2, \quad \forall v \in \hat{\mathbf{V}}_N^k.$$

Then we arrive at the following error estimate.

Theorem 3.4 (Error estimate in energy norm for sparse grid IPDG method) *Let u be the solution of (1.1)–(1.2) satisfying $u \in L^\infty(0, T; \mathcal{H}^{p+1}(\Omega))$, $u_t \in L^\infty(0, T; \mathcal{H}^{p+1}(\Omega))$, $u_{tt} \in L^1(0, T; \mathcal{H}^p(\Omega))$. u_h is the semi-discrete DG solution obtained by (3.1)–(3.3) with $\mathbf{V} = \hat{\mathbf{V}}_N^k$ and the initial condition $u_h(0) = \mathbf{P}u_0$ and $(u_h)_t(0) = \mathbf{P}v_0$, where \mathbf{P} denotes the L^2 projection of a function onto the space $\hat{\mathbf{V}}_N^k$. Then for $k \geq 1$ and any $1 \leq q \leq \min\{p, k\}$, the error $e = u_h - u$ satisfies the estimation*

$$\begin{aligned} \|e_t\|_{L^\infty(0, T; L^2(\Omega))} + \sup_{t \in [0, T]} \||e_t\|| \leq C(\||e_t(0)\|| + \||e(0)\||) \\ + C \left| \log_2 h_N \right|^d h_N^q \left(|u|_{L^\infty(0, T; \mathcal{H}^{q+1}(\Omega))} + T |u_t|_{L^\infty(0, T; \mathcal{H}^{q+1}(\Omega))} + |u_{tt}|_{L^\infty(0, T; \mathcal{H}^q(\Omega))} \right) \end{aligned} \quad (3.8)$$

where the dimension $d \geq 2$. $|\cdot|_{\mathcal{H}^{q+1}(\Omega)}$ denotes mixed derivative norm of a function and was defined in [21]. Here and below, C denotes a generic constant that does not depend on h_N or the solution u .

Proof Following [20], we let $\mathbf{V}(h) = H^1(\Omega) + \hat{\mathbf{V}}_N^k$, and for any $v \in \mathbf{V}(h)$, we define the lifted function $\mathcal{L}_c(v) \in (\hat{\mathbf{V}}_N^k)^d$ by requiring

$$\int_{\Omega} \mathcal{L}_c(v) \cdot w d\mathbf{x} = \sum_{e \in \Gamma} \int_e [v] \cdot \{c^2 w\} ds, \quad w \in (\hat{\mathbf{V}}_N^k)^d. \quad (3.9)$$

Using similar arguments as in Lemma 4.3 in [20], we conclude the lifting operator \mathcal{L}_c exists and is stable in the DG norm. Then the auxiliary bilinear form can be introduced as

$$\begin{aligned} \hat{B}(u, v) = \int_{\Omega} c^2 \nabla u \cdot \nabla v d\mathbf{x} - \int_{\Omega} \mathcal{L}_c(u) \cdot \nabla v ds - \int_{\Omega} \mathcal{L}_c(v) \cdot \nabla u ds \\ + \sum_{e \in \Gamma} \frac{\sigma}{h_N} \int_e [u] \cdot [v] ds. \end{aligned} \quad (3.10)$$

$\hat{B}(u, v)$ can be viewed as an extension of the wave operator and bilinear form $B(u, v)$ to the space $\mathbf{V}(h) \times \mathbf{V}(h)$, since

$$\hat{B}(u, v) = B(u, v) \quad \text{on } \hat{\mathbf{V}}_N^k \times \hat{\mathbf{V}}_N^k, \quad (3.11)$$

$$\hat{B}(u, v) = \int_{\Omega} c^2 \nabla u \cdot \nabla v d\mathbf{x} - \int_{\partial\Omega} (c^2 \nabla u) v \cdot \mathbf{n} ds \quad \text{on } H^1(\Omega) \times H^1(\Omega). \quad (3.12)$$

Moreover, it can be verified that

$$\begin{aligned} \hat{B}(u, v) &\leq C_b \||u\|| \cdot \||v\||, \\ \hat{B}(u, u) &\geq C_s \||u\||^2. \end{aligned} \quad (3.13)$$

Similar to Lemma 4.5 in [20], e satisfies the equation

$$(e_{tt}, v) + \hat{B}(e, v) = r_h(u, v), \quad \forall v \in \hat{\mathbf{V}}_N^k \quad (3.14)$$

where

$$r_h(u, v) = \sum_{e \in \Gamma} \int_e [v] \cdot \{c^2 \nabla u - c^2 \mathbf{P}(\nabla u)\} ds. \quad (3.15)$$

Therefore, we will have

$$\begin{aligned} \frac{1}{2} \frac{d}{dt} [||e_t||^2 + \hat{B}(e, e)] &= (e_{tt}, e_t) + \hat{B}(e, e_t) \\ &= (e_{tt}, (u - \mathbf{P}u)_t) + \hat{B}(e, (u - \mathbf{P}u)_t) + r_h(u, (\mathbf{P}u - u_h)_t). \end{aligned} \quad (3.16)$$

Integrating (3.16) over $[0, s]$ for any $s \in [0, T]$ yields

$$\begin{aligned} \frac{1}{2} ||e_t(s)||^2 + \frac{1}{2} \hat{B}(e(s), e(s)) &= \frac{1}{2} ||e_t(0)||^2 + \frac{1}{2} \hat{B}(e(0), e(0)) + \int_0^s (e_{tt}, (u - \mathbf{P}u)_t) dt \\ &\quad + \int_0^s \hat{B}(e, (u - \mathbf{P}u)_t) dt + \int_0^s r_h(u, (\mathbf{P}u - u_h)_t) dt. \end{aligned} \quad (3.17)$$

Because

$$\int_0^s (e_{tt}, (u - \mathbf{P}u)_t) dt = - \int_0^s (e_t, (u - \mathbf{P}u)_{tt}) dt + [(e_t, (u - \mathbf{P}u)_t)]_{t=0}^{t=s}, \quad (3.18)$$

and the inequalities (3.13) hold, together with Holder's inequalities, we will have

$$\begin{aligned} \frac{1}{2} ||e_t(s)||^2 + \frac{1}{2} C_s |||e(s)|||^2 &\leq \frac{1}{2} ||e_t(0)||^2 + \frac{1}{2} C_b |||e(0)|||^2 \\ &\quad + ||e_t||_{L^\infty(0, T; L^2(\Omega))} (|||u - \mathbf{P}u)_{tt}||_{L^1(0, T; L^2(\Omega))} \\ &\quad + 2 |||u - \mathbf{P}u)_t||_{L^\infty(0, T; L^2(\Omega))} \\ &\quad + C_b T |||e||| \cdot |||u - \mathbf{P}u)_t||| \\ &\quad + \left| \int_0^T r_h(u, (\mathbf{P}u - u_h)_t) dt \right|. \end{aligned} \quad (3.19)$$

Since the inequality (3.19) holds for any $s \in [0, T]$, taking the maximum on $[0, T]$ will result in

$$||e_t||_{L^\infty(0, T; L^2(\Omega))}^2 + C_s |||e|||_{L^\infty(0, T; \mathbf{V}(h))}^2 \leq ||e_t(0)||^2 + C_b |||e(0)|||^2 + T_1 + T_2 + T_3 \quad (3.20)$$

where the short-hand notation $|||e|||_{L^\infty(0, T; \mathbf{V}(h))} := \sup_{t \in [0, T]} |||e|||$ is introduced, and

$$\begin{aligned} T_1 &= 2 |||e_t|||_{L^\infty(0, T; L^2(\Omega))} (|||u - \mathbf{P}u)_{tt}|||_{L^1(0, T; L^2(\Omega))} + 2 |||u - \mathbf{P}u)_t|||_{L^\infty(0, T; L^2(\Omega))} \\ &\quad (3.21) \end{aligned}$$

$$T_2 = 2 C_b T |||e||| \cdot |||u - \mathbf{P}u)_t|||$$

$$T_3 = 2 \left| \int_0^T r_h(u, (\mathbf{P}u - u_h)_t) dt \right|.$$

Using the geometric-arithmetic mean inequality, and Lemma 3.2 in [21], we conclude

$$T_1 \leq \frac{1}{2} |||e_t|||_{L^\infty(0, T; L^2(\Omega))}^2 + 2 (|||u - \mathbf{P}u)_{tt}|||_{L^1(0, T; L^2(\Omega))}^2 + 2 |||u - \mathbf{P}u)_t|||_{L^\infty(0, T; L^2(\Omega))}^2 \quad (3.22)$$

$$\begin{aligned} &\leq \frac{1}{2} |||e_t|||_{L^\infty(0, T; L^2(\Omega))}^2 + 4 |||u - \mathbf{P}u)_{tt}|||_{L^1(0, T; L^2(\Omega))}^2 + 16 |||u - \mathbf{P}u)_t|||_{L^\infty(0, T; L^2(\Omega))}^2 \\ &\leq \frac{1}{2} |||e_t|||_{L^\infty(0, T; L^2(\Omega))}^2 + C \left| \log_2 h_N \right|^{2d} h_N^{2q} (|||u_{tt}|||_{L^\infty(0, T; \mathcal{H}^q(\Omega))}^2 + h_N^2 |||u_t|||_{L^2(0, T; \mathcal{H}^{q+1}(\Omega))}^2). \end{aligned}$$

Similarly

$$\begin{aligned} T_2 &\leq \frac{1}{4} C_s ||e||^2 + 4 \frac{C_b^2}{C_s} T^2 ||(u - \mathbf{P}u)_t||^2 \\ &\leq \frac{1}{4} C_s ||e||_{L^\infty(0, T; \mathbf{V}(h))}^2 + CT^2 |\log_2 h_N|^{2d} h_N^{2q+2} |u_t|_{L^2(0, T; \mathcal{H}^{q+1}(\Omega))}^2. \end{aligned} \quad (3.23)$$

We then start to bound the term T_3 . From (3.15), we can derive

$$\begin{aligned} |r_h(u, v)| &= \left| \sum_{e \in \Gamma} \int_e [v] \cdot \{c^2 \nabla u - c^2 \mathbf{P}(\nabla u)\} ds \right| \\ &\leq \left(\sum_{e \in \Gamma} \int_e \frac{\sigma}{h_N} [v]^2 ds \right)^{\frac{1}{2}} \left(\sum_{e \in \Gamma} \int_e \frac{h_N}{\sigma} |c^2 \nabla u - c^2 \mathbf{P}(\nabla u)|^2 ds \right)^{\frac{1}{2}} \\ &\leq C ||v|| \left(\sum_{K \in \Omega_N} h_N ||\nabla u - \mathbf{P}(\nabla u)||_{\partial K}^2 \right)^{\frac{1}{2}}, \end{aligned} \quad (3.24)$$

with $h_N = \frac{1}{2^N}$, using trace inequality and Lemma 3.2 in [21], we have

$$|r_h(u, v)| \leq C ||v||_{L^\infty(0, T; \mathbf{V}(h))} \cdot |\log_2 h_N|^d h_N^q |u|_{L^\infty(0, T; \mathcal{H}^{q+1}(\Omega))}. \quad (3.25)$$

Therefore,

$$\begin{aligned} \left| \int_0^T r_h(u, v_t) dt \right| &= \left| - \int_0^T r_h(u_t, v) dt + r_h(u, v)|_{t=0}^T \right| \\ &\leq CT ||v||_{L^\infty(0, T; \mathbf{V}(h))} |\log_2 h_N|^d h_N^q |u_t|_{L^\infty(0, T; \mathcal{H}^{q+1}(\Omega))} \\ &\quad + 2C ||v||_{L^\infty(0, T; \mathbf{V}(h))} |\log_2 h_N|^d h_N^q |u|_{L^\infty(0, T; \mathcal{H}^{q+1}(\Omega))}. \end{aligned} \quad (3.26)$$

Denote $\mathcal{R} = T |u_t|_{L^\infty(0, T; \mathcal{H}^{q+1}(\Omega))} + 2 |u|_{L^\infty(0, T; \mathcal{H}^{q+1}(\Omega))}$, we will have

$$\begin{aligned} T_3 &\leq 2C\mathcal{R} |\log_2 h_N|^d h_N^q ||\mathbf{P}u - u_h||_{L^\infty(0, T; \mathbf{V}(h))} \\ &\leq 2C\mathcal{R} |\log_2 h_N|^d h_N^q \left[||e||_{L^\infty(0, T; \mathbf{V}(h))} + ||u - \mathbf{P}u||_{L^\infty(0, T; \mathbf{V}(h))} \right] \\ &\leq \frac{1}{4} C_s ||e||_{L^\infty(0, T; \mathbf{V}(h))}^2 + C |\log_2 h_N|^{2d} h_N^{2q} \left[|u|_{L^\infty(0, T; \mathcal{H}^{q+1}(\Omega))}^2 + \mathcal{R}^2 \right]. \end{aligned} \quad (3.27)$$

Together with (3.20) and the estimates for T_1 , T_2 and T_3 , we arrive at the estimate

$$\begin{aligned} \frac{1}{2} ||e_t||_\infty^2 + \frac{1}{2} C_s \sup_{t \in [0, T]} ||e||^2 &\leq ||e_t(0)||^2 + C ||e(0)||^2 \\ &\quad + C |\log_2 h_N|^{2d} h_N^{2q} (|u_{tt}|_{L^\infty(0, T; \mathcal{H}^q(\Omega))}^2 + T^2 |u_t|_{L^\infty(0, T; \mathcal{H}^{q+1}(\Omega))}^2 \\ &\quad + |u|_{L^\infty(0, T; \mathcal{H}^{q+1}(\Omega))}^2), \end{aligned} \quad (3.28)$$

and this completes the proof.

3.2 Semi-Discrete Scheme with Multiresolution Interpolation

To treat variable coefficient case, we follow the idea in [27, 34] and interpolate the functions $c^2 u_h$ and $c^2 \nabla u_h$ (or $(c^2)^- \nabla u_h$ and $(c^2)^+ \nabla u_h$ in the case when $c^2(\mathbf{x})$ contains discontinuity on the cell interfaces of Ω_N) by using the multiresolution Lagrange interpolation discussed in Sect. 2.2. For simplicity of discussion, we only focus on the homogeneous Dirichlet

boundary condition with no source term. However, similar results can be established for mixed boundary conditions and also with source terms.

We first assume $c = c(\mathbf{x})$ is continuous. In this case, we can reformulate (3.1) into an equivalent form

$$\begin{aligned} B(u_h, v) &= \int_{\Omega} c^2 \nabla u_h \cdot \nabla v \, d\mathbf{x} - \sum_{e \in \Gamma} \int_e \{c^2 \nabla u_h\} \cdot [v] \, ds - \sum_{e \in \Gamma} \int_e \{\nabla v\} \cdot [c^2 u_h] \, ds \\ &\quad + \sum_{e \in \Gamma} \frac{\sigma}{h_N} \int_e [u_h] \cdot [v] \, ds \end{aligned} \quad (3.29)$$

then the scheme is implemented by the modified operator with interpolation

$$\begin{aligned} \tilde{B}(u_h, v) &= \int_{\Omega} \mathcal{I}_h(c^2 \nabla u_h) \cdot \nabla v \, d\mathbf{x} - \sum_{e \in \Gamma} \int_e \{\mathcal{I}_h(c^2 \nabla u_h)\} \cdot [v] \, ds - \sum_{e \in \Gamma} \int_e \{\nabla v\} \cdot [\mathcal{I}_h(c^2 u_h)] \, ds \\ &\quad + \sum_{e \in \Gamma} \frac{\sigma}{h_N} \int_e [u_h] \cdot [v] \, ds \end{aligned} \quad (3.30)$$

Here, $\mathcal{I}_h(\cdot)$ denote the interpolation operator defined in Sect. 2.2 with interpolation parameter M to be specified later.

If $c = c(\mathbf{x})$ is discontinuous along the cell interface, then some special care has to be taken for the third term $\sum_{e \in \Gamma} \int_e \{c^2 \nabla v\} \cdot [u_h] \, ds$. We first reformulate it into another form:

$$\begin{aligned} &\{c^2 \nabla v\} \cdot [u_h] \\ &= \frac{1}{2} ((c^2 \nabla v)^- + (c^2 \nabla v)^+) \cdot (u_h^- \mathbf{n}^- + u_h^+ \mathbf{n}^+) \\ &= \frac{1}{2} ((c^2 \nabla v)^- \cdot u_h^- \mathbf{n}^- + (c^2 \nabla v)^- \cdot u_h^+ \mathbf{n}^+ + (c^2 \nabla v)^+ \cdot u_h^- \mathbf{n}^- + (c^2 \nabla v)^+ \cdot u_h^+ \mathbf{n}^+) \\ &= \frac{1}{2} ((c^2)^- u_h^- \mathbf{n}^- \cdot (\nabla v)^- + (c^2)^- u_h^+ \mathbf{n}^+ \cdot (\nabla v)^- + (c^2)^+ u_h^- \mathbf{n}^- \cdot (\nabla v)^+ + (c^2)^+ u_h^+ \mathbf{n}^+ \cdot (\nabla v)^+) \\ &= \frac{1}{2} ((c^2)^- u_h^- \mathbf{n}^- + (c^2)^- u_h^+ \mathbf{n}^+) \cdot (\nabla v)^- + \frac{1}{2} ((c^2)^+ u_h^- \mathbf{n}^- + (c^2)^+ u_h^+ \mathbf{n}^+) \cdot (\nabla v)^+ \\ &= \frac{1}{2} [(c^2)^- u_h] \cdot (\nabla v)^- + \frac{1}{2} [(c^2)^+ u_h] \cdot (\nabla v)^+. \end{aligned}$$

Here $[(c^2)^- u_h] := ((c^2)^- u_h^- \mathbf{n}^- + (c^2)^- u_h^+ \mathbf{n}^+)$ and $[(c^2)^+ u_h] := ((c^2)^+ u_h^- \mathbf{n}^- + (c^2)^+ u_h^+ \mathbf{n}^+)$. Now the bilinear form (3.1) is rewritten into

$$\begin{aligned} B(u_h, v) &= \int_{\Omega} c^2 \nabla u_h \cdot \nabla v \, d\mathbf{x} - \sum_{e \in \Gamma} \int_e \{c^2 \nabla u_h\} \cdot [v] \, ds \\ &\quad - \sum_{e \in \Gamma} \int_e \left(\frac{1}{2} [(c^2)^- u_h] \cdot (\nabla v)^- + \frac{1}{2} [(c^2)^+ u_h] \cdot (\nabla v)^+ \right) \, ds + \sum_{e \in \Gamma} \frac{\sigma}{h_N} \int_e [u_h] \cdot [v] \, ds \end{aligned}$$

and then the interpolation operator is performed on $c^2 \nabla u_h$, $(c^2)^- u_h$ and also $(c^2)^+ u_h$, which gives:

$$\begin{aligned} \tilde{B}(u_h, v) &= \int_{\Omega} \mathcal{I}_h(c^2 \nabla u_h) \cdot \nabla v \, d\mathbf{x} - \sum_{e \in \Gamma} \int_e \{\mathcal{I}_h(c^2 \nabla u_h)\} \cdot [v] \, ds \\ &\quad - \sum_{e \in \Gamma} \int_e \left(\frac{1}{2} [\mathcal{I}_h((c^2)^- u_h)] \cdot (\nabla v)^- + \frac{1}{2} [\mathcal{I}_h((c^2)^+ u_h)] \cdot (\nabla v)^+ \right) \, ds + \sum_{e \in \Gamma} \frac{\sigma}{h_N} \int_e [u_h] \cdot [v] \, ds. \end{aligned} \quad (3.31)$$

Remark 3.5 One can also try to perform the interpolation on the coefficient c^2 . Actually, in most applications, c^2 only depends on x and does not change with time t . In this case, we only

need to interpolate c^2 once and thus save much computational cost. However, the evaluation of the residual will be much more complicated, since there are two vectors of coefficients here (the coefficients of c^2 with the interpolation basis and ∇u_h with the gradient of Alpert's basis) in the residual. The fast matrix-vector multiplication in Sect. 3.4 cannot be directly applied here and some new fast algorithm needs to be used.

Following [14], we can now write the DG scheme with interpolation (3.31) into the semi-discrete form as

$$\frac{d^2 u_h}{dt^2} = L_h(u_h), \quad (3.32)$$

where $L_h(u)$ is an operator onto \mathbf{V} which is a discrete approximation of $-\nabla \cdot (c^2(\mathbf{x}) \nabla u)$ and satisfies

$$\begin{aligned} \sum_{K \in \Omega_N} \int_K L_h(u_h) v_h \, d\mathbf{x} &= - \int_{\Omega} \mathcal{I}_h(c^2 \nabla u_h) \cdot \nabla v_h \, d\mathbf{x} + \sum_{e \in \Gamma} \int_e \{\mathcal{I}_h(c^2 \nabla u_h)\} \cdot [v_h] \, ds \\ &\quad + \sum_{e \in \Gamma} \int_e \left(\frac{1}{2} [\mathcal{I}_h((c^2)^- u_h)] \cdot (\nabla v_h)^- + \frac{1}{2} [\mathcal{I}_h((c^2)^+ u_h)] \cdot (\nabla v_h)^+ \right) \, ds \\ &\quad - \sum_{e \in \Gamma} \frac{\sigma}{h_N} \int_e [u_h] \cdot [v_h] \, ds \end{aligned} \quad (3.33)$$

for any $v_h \in \mathbf{V}$.

To preserve the accuracy of the original DG scheme, interpolation operator $\mathcal{I}_h(\cdot)$ needs to reach certain accuracy. Using similar techniques as in [14, 28], we have the following proposition on local truncation error of the sparse grid method with $\mathbf{V} = \hat{\mathbf{V}}_N^k$. We only discuss the case when $c(\mathbf{x})$ is discontinuous, since the similar approach can be applied when $c(\mathbf{x})$ is continuous.

Proposition 3.6 (Local truncation error analysis) *If the interpolation operator \mathcal{I}_h in (3.31) has the accuracy of order $|\log_2 h_N|^d h_N^{k+3}$ for sufficiently smooth functions, then the local truncation error of the semi-discrete DG scheme with interpolation (3.31) is of order $|\log_2 h_N|^d h_N^{k+1}$. To be more precise, for sufficiently smooth function u , the sparse grid DG method with interpolation (3.31) has the truncation error:*

$$\|L_h(u) + \nabla \cdot (c^2(\mathbf{x}) \nabla u)\|_{L^2(\Omega)} \leq C |\log_2 h_N|^d h_N^{k+1}. \quad (3.34)$$

Here, we use C to denote any generic constant that may depend on the solution u and $c(\mathbf{x})$, but does not depend on N .

Proof We denote the standard L^2 projection operator onto the sparse grid DG finite element space by \mathbf{P} , then

$$\|L_h(u) + \nabla \cdot (c^2(\mathbf{x}) \nabla u)\| \leq e_1 + e_2, \quad (3.35)$$

where

$$e_1 := \|L_h(u) + \mathbf{P}(\nabla \cdot (c^2(\mathbf{x}) \nabla u))\|,$$

and

$$e_2 := \|\mathbf{P}(\nabla \cdot (c^2(\mathbf{x}) \nabla u)) - \nabla \cdot (c^2(\mathbf{x}) \nabla u)\|$$

The estimate for e_2 can be obtained by projection properties [21]:

$$e_2 \leq C |\log_2 h_N|^d h_N^{k+1}. \quad (3.36)$$

To estimate e_1 , we consider any test function v_h in DG space, and obtain

$$\begin{aligned} & \sum_{K \in \Omega_N} \int_K (L_h(u) + \mathbf{P}(\nabla \cdot (c^2(\mathbf{x}) \nabla u)) v_h) d\mathbf{x} = \sum_{K \in \Omega_N} \int_K (L_h(u) + \nabla \cdot (c^2(\mathbf{x}) \nabla u)) v_h d\mathbf{x} \\ &= - \int_{\Omega} \mathcal{I}_h(c^2 \nabla u) \cdot \nabla v_h d\mathbf{x} + \sum_{e \in \Gamma} \int_e \{\mathcal{I}_h(c^2 \nabla u)\} \cdot [v_h] ds \\ & \quad + \sum_{e \in \Gamma} \int_e \left(\frac{1}{2} [\mathcal{I}_h((c^2)^- u)] \cdot (\nabla v_h)^- + \frac{1}{2} [\mathcal{I}_h((c^2)^+ u)] \cdot (\nabla v_h)^+ \right) ds - \sum_{e \in \Gamma} \frac{\sigma}{h_N} \int_e [u] \cdot [v_h] ds \\ & \quad + \int_{\Omega} c^2 \nabla u \cdot \nabla v_h d\mathbf{x} - \sum_{e \in \Gamma} \int_e \{c^2 \nabla u\} \cdot [v_h] ds \\ & \quad - \sum_{e \in \Gamma} \int_e \left(\frac{1}{2} [(c^2)^- u] \cdot (\nabla v_h)^- + \frac{1}{2} [(c^2)^+ u] \cdot (\nabla v_h)^+ \right) ds + \sum_{e \in \Gamma} \frac{\sigma}{h_N} \int_e [u] \cdot [v_h] ds \\ &= - \int_{\Omega} (\mathcal{I}_h(c^2 \nabla u) - c^2 \nabla u) \cdot \nabla v_h d\mathbf{x} + \sum_{e \in \Gamma} \int_e \{\mathcal{I}_h(c^2 \nabla u) - c^2 \nabla u\} \cdot [v_h] ds \\ & \quad + \sum_{e \in \Gamma} \int_e \left(\frac{1}{2} [\mathcal{I}_h((c^2)^- u) - (c^2)^- u] \cdot (\nabla v_h)^- + \frac{1}{2} [\mathcal{I}_h((c^2)^+ u) - (c^2)^+ u] \cdot (\nabla v_h)^+ \right) ds \\ & \leq \|\mathcal{I}_h[c^2 \nabla u] - c^2 \nabla u\| \|\nabla v_h\| + \|\mathcal{I}_h[c^2 \nabla u] - c^2 \nabla u\|_{L^2(\Gamma_h)} \|v_h\|_{L^2(\Gamma_h)} \\ & \quad + \|\mathcal{I}_h[c^2 \nabla u] - c^2 \nabla u\|_{L^2(\Gamma_h)} \|\nabla v_h\|_{L^2(\Gamma_h)} \\ & \leq C |\log_2 h_N|^d h_N^{k+3} h_N^{-1} \|v_h\| + C h_N^{-\frac{1}{2}} |\log_2 h_N|^d h_N^{k+3} h_N^{-\frac{1}{2}} \|v_h\| + C h_N^{-\frac{1}{2}} |\log_2 h_N|^d h_N^{k+3} h_N^{-\frac{3}{2}} \|v_h\| \\ & = C |\log_2 h_N|^d h_N^{k+1} \|v_h\|. \end{aligned}$$

Here, we have used the multiplicative trace inequality and the inverse inequality, see e.g. Lemma 2.1 and Lemma 2.3 in [28]. By taking $v_h = (L_h(u) + \mathbf{P}(\nabla \cdot (c^2(\mathbf{x}) \nabla u)))$ in the inequality above, we have

$$e_1 = \|L_h(u) + \mathbf{P}(\nabla \cdot (c^2(\mathbf{x}) \nabla u))\| \leq C |\log_2 h_N|^d h_N^{k+1}. \quad (3.37)$$

Combining (3.37) and (3.36), we have the estimate for the truncation error (3.34).

Remark 3.7 The proposition above indicates that, to preserve the order of the original scheme, we should use $M \geq k + 2$. For example, if we take piecewise linear polynomials for the DG space, then it is required to apply cubic interpolation operator to treat the nonlinear terms. From our numerical tests, however, this seems that it is not a necessary condition. To reach the desired convergence rate, one only needs to take $M \geq k$.

In Proposition 3.6, we only estimate the local truncation error, and this is far from a rigorous error estimate that takes into account stability. Unlike the scheme with the symmetric bilinear form $B(u_h, v)$ as in Theorem 3.1, the symmetry is lost in the interpolated bilinear form $\tilde{B}(u_h, v)$. Hence, energy stability is not automatic. In numerical experiments, we observe that the sparse grid DG method with Lagrange interpolation with only inner interpolation points is unstable for polynomials of high degrees (see the numerical results in Table 5 in Sect. 4). With the interpolation points at the *interface*, the sparse grid DG scheme is stable and yields satisfactory convergence rate (see Table 6 in Sect. 4).

3.3 Time Stepping and Adaptivity

For time discretizations, we first write the second order semi-discrete scheme (3.31)

$$(u_h)_{tt} = L_h(u_h) \quad (3.38)$$

into a first order system

$$\begin{aligned} (u_h)_t &= w_h, \\ (w_h)_t &= L_h(u_h), \end{aligned}$$

and then apply the standard Runge–Kutta scheme. The reason why we use the one-step RK method instead of the multistep method is that the maximum allowed time step size from the CFL restriction may change with the adaptive mesh in different time steps. This would result in additional computational cost in extrapolation or interpolation between different time steps for the multistep methods. We remark that other types of one step method can also be used but is not pursued in this work.

The adaptive scheme uses the procedure developed in [8,22] to determine the space \mathbf{V} that dynamically evolves over time. The method is very similar to those in [8,22], and the details are omitted for brevity. The main difference is that we need to keep track of two sets of basis functions corresponding to the same adaptive space are involved [27]. Another difference is that the refinement and the coarsening criteria are determined by the L^2 norms of both u_h and w_h , which are both important for predicting solution profiles for wave equations. There are some cases which start with a zero displacement u but a non-zero velocity u_t . If we only take the norms of u_h as an indicator, the adaptive procedure will result in poor resolutions. Only by considering the norms of both u_h and w_h , one can capture the profiles well.

3.4 Fast Algorithms

We now describe the fast matrix-vector multiplication algorithm, which is essential for efficient implementation of our schemes. Because the multiwavelet bases are global, the evaluation of the residual yields denser matrix than those obtained by standard local bases. Efficient implementations are therefore essential to ensure that the computational cost is on par with element-wise implementation of traditional DG schemes. This issue has been also discussed in our work for conservation laws [27], which extends the fast matrix-vector multiplication in [34,42] to adaptive index set.

Following [27,34], we consider matrix-vector multiplication in multi-dimensions in an abstract framework.

$$f_{\mathbf{n}} = \sum_{H(\mathbf{n}') \leq 0} f'_{\mathbf{n}'} t_{n'_1, n_1}^{(1)} t_{n'_2, n_2}^{(2)} \cdots t_{n'_d, n_d}^{(d)}, \quad H(\mathbf{n}) \leq 0, \quad (3.39)$$

where $\mathbf{n} = (n_1, n_2, \dots, n_d)$ and $\mathbf{n}' = (n'_1, n'_2, \dots, n'_d)$ can be thought of as the level of the mesh, and $t_{n'_1, n_1}^{(i)} = T_{n'_1, n_1}^{(i)}$ represents the calculations in the i -th dimension. It is assumed that the constraint function $H = H(\mathbf{n}') = H(n'_1, n'_2, \dots, n'_d)$ is non-decreasing with respect to each variable. This holds true for sparse grid (by taking $H(\mathbf{n}') = |\mathbf{n}'|_1 - N$) and also for adaptive multiresolution method.

One can compute the sum (3.39) dimension-by-dimension, i.e. we first perform the transformation in the x_1 dimension:

$$g_{(n_1, n'_2, \dots, n'_d)}^{(1)} = \sum_{H(n'_1, n'_2, \dots, n'_d) \leq 0} f'_{(n'_1, n'_2, \dots, n'_d)} t_{n'_1, n_1}^{(1)}, \quad (3.40)$$

and then in the x_2 dimension:

$$g_{(n_1, n_2, n'_3, \dots, n'_d)}^{(2)} = \sum_{H(n_1, n'_2, \dots, n'_d) \leq 0} g_{(n_1, n'_2, \dots, n'_d)}^{(1)} t_{n'_2, n_2}^{(2)}, \quad (3.41)$$

and all the way up to x_d dimension:

$$f_{(n_1, n_2, n_3, \dots, n_d)} = \sum_{H(n_1, n_2, \dots, n_{d-1}, n'_d) \leq 0} g_{(n_1, n_2, \dots, n_{d-1}, n'_d)}^{(d-1)} t_{n'_d, n_d}^{(d)}. \quad (3.42)$$

It can be proved that (3.40)-(3.42) is equivalent to the original summation (3.39), if assuming that, for some integer $1 \leq k \leq d$, $T^{(i)}$ for $i = 1, \dots, k-1$ are strictly lower triangular and $T^{(i)}$ for $i = k+1, \dots, d$ are upper triangular (or $T^{(i)}$ for $i = 1, \dots, k-1$ are lower triangular and $T^{(i)}$ for $i = k+1, \dots, d$ are strictly upper triangular) [34]. Here, $T^{(i)}$ denotes the i -th transformation matrix. When such properties for $T^{(i)}$ matrices are not true, one can perform $L + U$ split and (3.39) becomes:

$$f_n = \sum_{H(n') \leq 0} f'_{n'} (l_{n'_1, n_1}^{(1)} + u_{n'_1, n_1}^{(1)}) (l_{n'_2, n_2}^{(2)} + u_{n'_2, n_2}^{(2)}) \cdots (l_{n'_{d-1}, n_{d-1}}^{(d-1)} + u_{n'_{d-1}, n_{d-1}}^{(d-1)}) t_{n'_d, n_d}^{(d)}, \quad (3.43)$$

where there are in total 2^{d-1} terms that can be computed dimension-by-dimension. The overall computational cost is $\mathcal{O}(2^{d-1} \cdot \text{DoF} \cdot N)$ if the cost of one-dimensional transform is log-linear, i.e., $\mathcal{O}(\mathcal{N} \log \mathcal{N})$ where \mathcal{N} denotes the DoF in one-dimension [34]. This assumption holds true for our sparse grid DG scheme.

We apply this fast matrix-vector multiplication in several parts of our algorithm. We will discuss the details about initialization, which is the procedure to project the initial value onto the DG finite element space represented by multiwavelet bases. When the given initial value is separable, i.e.,

$$u(x) = \prod_{i=1}^d u_i(x_i), \quad (3.44)$$

one just need to project each 1D function $u_i = u_i(x_i)$ for $i = 1, \dots, d$ onto 1D multiwavelet bases and then we can easily get the projection of $u(x)$. This approach naturally extends to the case when the initial value is a summation of separable functions:

$$u(x) = \sum_{j=1}^n \left(\prod_{i=1}^d u_{j,i}(x_i) \right). \quad (3.45)$$

However, if the function is non-separable, direct evaluation of L^2 projection would result in very large computational cost if using numerical quadratures in multi-dimensions. Rather, we propose to apply the adaptive multiresolution interpolation introduced in [37] with appropriate error tolerance and approximate the solution using a collection of interpolation basis functions:

$$u_h(\mathbf{x}) = \sum_{\substack{(\mathbf{l}, \mathbf{j}) \in G, \\ \mathbf{l} \leq \mathbf{k} \leq \mathbf{k} + \mathbf{1}}} b_{\mathbf{l}, \mathbf{l}}^{\mathbf{j}} \psi_{\mathbf{l}, \mathbf{l}}^{\mathbf{j}}(\mathbf{x}) \quad (3.46)$$

with G the index of all active elements. Next we use the fast matrix-vector multiplication (3.39) to transform coefficients of interpolation basis $\{b_{i,1}^j\}$ to coefficients of Alpert's basis $\{c_{i,1}^j\}$:

$$u_h(\mathbf{x}) = \sum_{\substack{(\mathbf{l}, \mathbf{j}) \in G, \\ 1 \leq i \leq k+1}} c_{i,1}^j v_{i,1}^j(\mathbf{x}) \quad (3.47)$$

In (3.39), f_n and $f_{n'}$ represent $\{b_{i,1}^j\}$ and $\{c_{i,1}^j\}$, respectively. The matrix $t_{n'_i, n_i}^{(i)}$ is the product of 1D interpolation basis and 1D Alpert's basis.

We apply similar approach in the multiresolution interpolation and the evaluation of the right hand side of the weak formulation (3.31). We refer readers to [27] for details.

4 Numerical Examples

In this section, we perform numerical experiments to validate the performance of our scheme. We consider 2D or 3D problems with computational domain being $[0, 1]^d$ with $d = 2, 3$. The CFL number is taken to be 0.1 in 2D and 0.05 in 3D. The penalty parameter σ is taken to be 10 in 2D and 30 in 3D, unless otherwise stated. For the accuracy test, we check the convergence order for P^k DG with $k = 1, 2, 3$ coupled with RK time discretization. In particular, for $k = 1, 2$, we use the second and the third-order strong stability preserving Runge–Kutta method [19, 35], and for $k = 3$, we use the classical RK4 methods. All adaptive calculations are obtained by $k = 3$ and RK4 time stepping. In the adaptive scheme, we take $\eta = \epsilon/10$. DoF=dim(\mathbf{V}^k) refers to the number of Alpert's multiwavelets basis functions in the adaptive grids. The maximum mesh level N is taken to be 8, unless otherwise stated.

Example 4.1 (wave equation with constant coefficient) In this example, consider the d -dimensional wave equation with a constant coefficient

$$u_{tt} = \sum_{i=1}^d u_{x_i x_i} \quad (4.1)$$

on the domain $[0, 1]^d$. We take the exact solution to be

$$u(\mathbf{x}, t) = \sin(a\sqrt{d}\pi t) \prod_{i=1}^d \cos(a\pi x_i)$$

with a being a constant and various types of boundary conditions.

- (a) We take $a = 2$ and $d = 2, 3$ with periodic boundary conditions.
- (b) We take $a = 1$ and $d = 2, 3$ and incorporate Dirichlet boundary condition in the x_1 -direction and Neumann boundary in other directions.

Note that in this example, (3.1) is implemented with no interpolation because c is a constant. To output the L^2 -error between the numerical solution u_h and the exact solution $u(x)$, we use the fact that

$$\int_{\Omega} (u_h - u)^2 dx = \int_{\Omega} u_h^2 dx - 2 \int_{\Omega} u_h u dx + \int_{\Omega} u^2 dx.$$

Table 1 Example 4.1(a): wave equation with constant coefficients, periodic boundary conditions, sparse grid DG, $k = 1, 2, 3, d = 2, 3, t = 0.1$

N	$k = 1$		N	$k = 2$		N	$k = 3$	
	L^2 -error	Order		L^2 -error	Order		L^2 -error	Order
$d = 2$								
5	5.90e-03	–	5	1.96e-04	–	3	2.80e-04	–
6	1.69e-03	1.81	6	3.03e-05	2.69	4	1.80e-05	3.96
7	4.66e-04	1.86	7	4.43e-06	2.77	5	1.48e-06	3.61
8	1.23e-04	1.92	8	6.21e-07	2.83	6	1.10e-07	3.76
$d = 3$								
5	1.58e-02	–	5	7.38e-04	–	3	4.50e-04	–
6	8.66e-03	0.87	6	1.68e-04	2.14	4	8.02e-05	2.49
7	2.42e-03	1.84	7	3.03e-05	2.47	5	5.24e-06	3.94
8	8.41e-04	1.53	8	5.30e-06	2.51	6	5.30e-07	3.30

Table 2 Example 4.1(b): wave equation with constant coefficients, Dirichlet and Neumann boundary conditions, sparse grid DG, $k = 1, 2, 3, d = 2, 3, t = 0.1$

N	$k = 1$		N	$k = 2$		N	$k = 3$	
	L^2 -error	Order		L^2 -error	Order		L^2 -error	Order
$d = 2$								
3	3.30e-03	–	3	1.21e-04	–	1	4.51e-04	–
4	1.21e-03	1.44	4	1.79e-05	2.75	2	6.34e-05	2.83
5	2.94e-04	2.04	5	2.43e-06	2.89	3	8.30e-06	2.93
6	8.15e-05	1.85	6	3.41e-07	2.83	4	7.64e-07	3.44
$d = 3$								
3	2.15e-02	–	3	2.14e-04	–	1	4.44e-04	–
4	7.06e-03	1.61	4	3.39e-05	2.66	2	3.66e-05	3.60
5	2.04e-03	1.79	5	5.13e-06	2.73	3	2.39e-06	3.94
6	5.27e-04	1.95	6	1.08e-06	2.24	4	1.23e-07	4.28

The first term $\int_{\Omega} u_h^2 dx$ can be easily computed with the aid of the orthonormality of the Alpert's basis functions. The second term $\int_{\Omega} u_h u dx$ can be computed by the same fast approach as the initial projection, which has been explained in detail in Sect. 3.4. The third term $\int_{\Omega} u^2 dx$ can be computed analytically.

The numerical results obtained by sparse grid DG method are presented in Table 1 for case (a) and in Table 2 for case (b). For both cases, the convergence order is slightly bigger than k but smaller than $k + 1$, which is higher than the predicted rate in Theorem 3.4, but similar to the results for linear transport equation in [21]. The numerical results with adaptive method are shown in Tables 3 and 4. Similar to [22], we measure the convergence rates with respect to DoF: R_{DoF} and $\epsilon : R_{\epsilon}$. We can clearly observe the effectiveness of the adaptive algorithm, i.e. R_{ϵ} is close to 1. The convergence order R_{DoF} is bigger than $\frac{k+1}{d}$, which is the rate obtained by an optimally convergent non-adaptive scheme.

Table 3 Example 4.1(a): wave equation with constant coefficients, periodic boundary conditions, adaptive sparse grid DG, $k = 3$

ϵ	DoF	L^2 -error	R_{DoF}	R_ϵ
$d = 2$				
1e-1	128	1.25e-3	—	—
1e-2	320	2.80e-4	1.63	0.65
1e-3	1088	2.61e-5	1.94	1.03
1e-4	1536	4.02e-6	5.43	0.81
$d = 3$				
1e-1	896	2.16e-3	—	—
1e-2	2432	4.74e-4	1.52	0.66
1e-3	5888	8.30e-5	1.97	0.76
1e-4	28,160	6.69e-6	1.61	1.09

Table 4 Example 4.1(b): wave equation with constant coefficients, Dirichlet and Neumann boundary conditions, adaptive sparse grid DG, $k = 3$

ϵ	DoF	L^2 -error	R_{DoF}	R_ϵ
$d = 2$				
1e-1	32	1.11e-3	—	—
1e-2	112	6.35e-5	2.29	1.24
1e-3	208	1.70e-5	2.13	0.57
1e-4	384	2.35e-6	3.22	0.86
$d = 3$				
1e-1	64	1.29e-3	—	—
1e-2	640	3.71e-5	1.54	1.54
1e-3	1280	2.41e-5	0.62	0.19
1e-4	2368	3.12e-6	3.32	0.89

Example 4.2 (wave equation with smooth variable coefficient) This example tests wave equation with smooth variable coefficient

$$u_{tt} - \nabla \cdot (c^2(\mathbf{x}) \nabla u) = f, \quad (4.2)$$

on the computational domain $[0, 1]^d$ with $d = 2, 3$ and periodic boundary conditions.

For 2D case, we take

$$c^2(x_1, x_2) = (\cos(2\pi x_1) \cos(2\pi x_2) + 2)/3, \quad (4.3)$$

and the corresponding source term $f = f(x_1, x_2, t)$ such that the exact solution is

$$u = \sin(\pi t) \sin(2\pi x_1) \cos(2\pi x_2). \quad (4.4)$$

For 3D case, we take

$$c^2(x_1, x_2, x_3) = (\sin(2\pi x_1) \sin(2\pi x_2) \cos(2\pi x_3) + 2)/3, \quad (4.5)$$

Table 5 Example 4.2: wave equation with smooth variable coefficients in 2D, sparse grid DG, Lagrange interpolation with inner interpolation points, $k = 1, 2, 3, t = 0.1$

N	$M = 1$		$M = 2$		$M = 3$	
	L^2 -error	Order	L^2 -error	Order	L^2 -error	Order
$k = 1$						
3	2.52e-02	–	2.55e-02	–	2.52e-02	–
4	1.68e-02	0.59	1.63e-02	0.64	1.63e-02	0.63
5	3.67e-03	2.19	3.37e-03	2.28	3.36e-03	2.28
6	9.43e-04	1.96	9.62e-04	1.81	8.33e-04	2.01
$k = 2$						
3	1.64e-02	–	3.95e-03	–	4.73e-02	–
4	1.03e-02	0.67	7.89e-04	2.32	1.46e-01	-1.63
5	3.57e-03	1.53	7.80e-04	0.02	3.17e+02	-11.08
6	1.13e-02	-1.66	1.60e-01	-7.68	6.27e+10	-27.56
$k = 3$						
3	5.28e-03	–	2.14e+03	–	5.13e+03	–
4	5.55e-02	-3.40	1.11e+10	-22.31	2.23e+09	-18.73
5	1.93e+03	-15.09	1.39e+24	-46.83	4.78e+21	-40.97
6	8.15e+20	-58.55	9.83e+56	-109.12	1.40e+51	-97.88

and the corresponding source term $f = f(x_1, x_2, x_3, t)$ such that the exact solution is

$$u = \sin(\pi t) \sin(2\pi x_1) \cos(2\pi x_2) \cos(2\pi x_3). \quad (4.6)$$

This problem needs to invoke the fast interpolation methods to handle the variable coefficient. We first compare different choices of interpolation points. We use the inner interpolation points in Table 5 for 2D. The interpolation points and the basis functions are listed in the Appendix. When $k = 1$, the convergence order seems satisfactory. However, for $k = 2, 3$, the scheme is unstable. If we use Lagrange interpolation with the interface points, one will observe good convergence rate for $M \geq k$, as shown in Table 6. We also find that the error is almost the same for $M = k + 1$ and $M = k + 2$, and both much smaller than $M = k$. Therefore, in applications, we recommend taking $M = k + 1$ for accuracy considerations. Notice that this is a more relaxed condition from what is indicated by the local truncation analysis Proposition 3.6. We also remark that for nonlinear conservation laws in [27], Lagrange interpolation is unstable even with interface points, and Hermite interpolation has to be employed. However, for all numerical examples in this paper for linear wave equations with variable coefficients, Lagrange interpolation with interface points yields a stable scheme, and we choose to use this instead of Hermite interpolation due to its easier implementation.

For 3D cases, to save space, we only show numerical results with interface interpolation points in Table 7, in which good convergence rate is also observed. The result using adaptive method with $k = 3$ and $M = 4$ are presented in Table 8 for both 2D and 3D, and the conclusions are similar to the constant coefficient case.

Example 4.3 (wave equation with discontinuous coefficients) In this example, we consider wave equation with discontinuous coefficients. The jump of the coefficient aligns with the cell interface on the fine mesh Ω_N .

Table 6 Example 4.2: wave equation with smooth variable coefficients in 2D, sparse grid DG, Lagrange interpolation with interface interpolation points, $k = 1, 2, 3, t = 0.1$

N	M = 2		M = 3		M = 4	
	L^2 -error	order	L^2 -error	order	L^2 -error	order
$k = 1$						
3	2.52e-02	–	2.52e-02	–	2.52e-02	–
4	1.65e-02	0.61	1.63e-02	0.63	1.63e-02	0.63
5	3.52e-03	2.23	3.36e-03	2.28	3.36e-03	2.28
6	9.52e-04	1.89	8.30e-04	2.02	8.27e-04	2.02
$k = 2$						
3	2.69e-03	–	2.08e-03	–	2.08e-03	–
4	5.24e-04	2.36	4.38e-04	2.25	4.37e-04	2.25
5	1.25e-04	2.07	7.58e-05	2.53	7.58e-05	2.53
6	1.64e-05	2.93	1.16e-05	2.71	1.16e-05	2.71
$k = 3$						
3	2.92e-04	–	9.28e-05	–	8.75e-05	–
4	2.66e-05	3.46	1.05e-05	3.15	1.03e-05	3.09
5	3.04e-06	3.13	7.80e-07	3.74	7.68e-07	3.74
6	1.83e-07	4.05	5.10e-08	3.94	5.03e-08	3.93

Table 7 Example 4.2: wave equation with smooth variable coefficients in 3D, sparse grid DG, Lagrange interpolation with interface interpolation points, $k = 1, 2, 3, t = 0.1$

N	M = 1		M = 2		M = 3	
	L^2 -error	Order	L^2 -error	order	L^2 -error	Order
$k = 1$						
3	1.17e-01	–	1.17e-01	–	1.17e-01	–
4	2.20e-02	2.41	2.20e-02	2.41	2.20e-02	2.41
5	1.74e-02	0.34	1.71e-02	0.36	1.71e-02	0.36
6	4.65e-03	1.90	4.52e-03	1.92	4.51e-03	1.92
$k = 2$						
4	2.96e-03	–	1.58e-03	–	1.58e-03	–
5	7.78e-04	1.93	3.28e-04	2.27	3.27e-04	2.27
6	2.93e-04	1.41	6.58e-05	2.32	6.58e-05	2.32
7	3.88e-05	2.92	1.15e-05	2.52	1.15e-05	2.52
$k = 3$						
3	8.96e-04	–	3.88e-04	–	3.17e-04	–
4	2.05e-04	2.13	3.58e-05	3.44	2.19e-05	3.85
5	4.87e-05	2.07	3.27e-06	3.45	3.01e-06	2.86
6	5.80e-06	3.07	2.55e-07	3.68	2.31e-07	3.71

Table 8 Example 4.2, wave equation with smooth variable coefficient, adaptive sparse grid DG, 2D and 3D. $k = 3$, $M = 4$, $t = 0.1$

ϵ	DoF	L^2 -error	R_{DoF}	R_ϵ
<i>d</i> = 2				
1e-1	96	1.66e-3	—	—
1e-2	224	3.03e-4	2.00	0.74
1e-3	672	2.78e-5	2.18	1.04
1e-4	1088	3.17e-6	4.50	0.94
<i>d</i> = 3				
1e-1	576	2.11e-3	—	—
1e-2	1152	5.26e-4	2.00	0.60
1e-3	3584	8.73e-5	1.58	0.78
1e-4	8704	1.26e-5	2.18	0.84

For 2D case, the domain $\Omega = [0, 1]^2$ is composed of two subdomains $\Omega_1 = [\frac{1}{4}, \frac{3}{4}] \times [0, 1]$ and $\Omega_2 = \Omega \setminus \Omega_1$. The coefficient c^2 is a constant in each subdomain:

$$c^2 = \begin{cases} 1, & \text{in } \Omega_1, \\ \frac{5}{37}, & \text{in } \Omega_2. \end{cases} \quad (4.7)$$

Periodic boundary conditions are imposed in both x_1 - and x_2 - directions. With this setup, the exact solution is a standing wave

$$u = \begin{cases} \sin(\sqrt{20}\pi t) \cos(4\pi x_1) \cos(2\pi x_2), & \text{in } \Omega_1, \\ \sin(\sqrt{20}\pi t) \cos(12\pi x_1) \cos(2\pi x_2), & \text{in } \Omega_2. \end{cases} \quad (4.8)$$

For 3D case, $\Omega_1 = [\frac{1}{4}, \frac{3}{4}] \times [0, 1] \times [0, 1]$ and $\Omega_2 = \Omega \setminus \Omega_1$

$$c^2 = \begin{cases} 1, & \text{in } \Omega_1, \\ \frac{3}{19}, & \text{in } \Omega_2. \end{cases} \quad (4.9)$$

Periodic boundary conditions are imposed in all directions. With this setup, the exact solution is a standing wave

$$u = \begin{cases} \sin(\sqrt{24}\pi t) \cos(4\pi x_1) \cos(2\pi x_2) \cos(2\pi x_3), & \text{in } \Omega_1, \\ \sin(\sqrt{24}\pi t) \cos(12\pi x_1) \cos(2\pi x_2) \cos(2\pi x_3), & \text{in } \Omega_2. \end{cases} \quad (4.10)$$

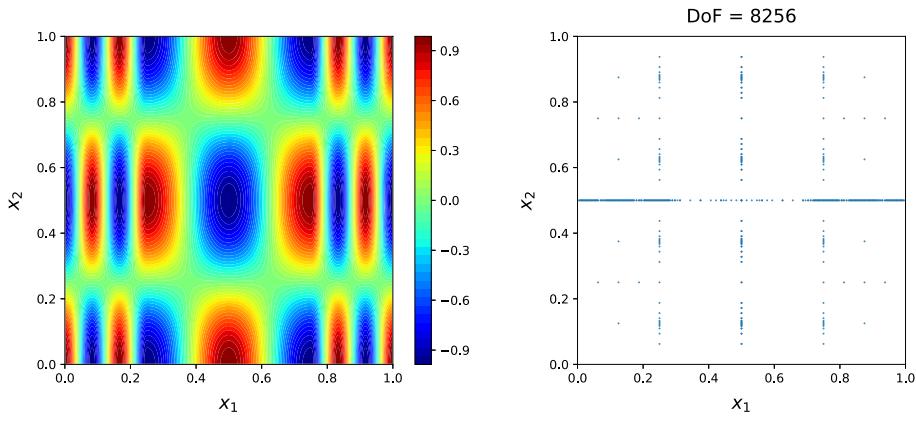
Since the solution is only piecewise smooth, the sparse grid DG method is not expected to have good convergence rate. Therefore, we only show the convergence result obtained by the adaptive method in Table 9 for both 2D and 3D. In addition, the adaptive result with the parameter $N = 8$ and $\epsilon = 1 \times 10^{-4}$ in 2D is shown in Fig. 1. There are fewer DoFs in the x_1 direction since the solution is smooth in that direction, and as expected, there are more DoFs located in the subdomain Ω_1 than that in $\Omega \setminus \Omega_1$.

Example 4.4 (Expanding wave in homogeneous medium) We consider the wave equation with constant wave speed $c = 1$ on the computational domain $\Omega = [0, 1]^d$. The homogeneous Neumann boundary conditions are used on all boundaries. The initial condition is taken as

$$u(\mathbf{x}, 0) = 0, \quad u_t(\mathbf{x}, 0) = 100e^{-500r^2} \quad (4.11)$$

Table 9 Example 4.3. discontinuous coefficient, adaptive sparse grid DG, 2D and 3D, $t = 0.01$

ϵ	DoF	L^2 -error	R_{DoF}	R_ϵ
<i>d</i> = 2				
1e-1	480	2.93e-4	—	—
1e-2	1088	8.43e-5	1.52	0.54
1e-3	2240	8.46e-6	3.18	1.00
1e-4	4224	1.05e-6	3.29	0.91
<i>d</i> = 3				
1e-1	2304	5.59e-4	—	—
1e-2	7040	1.28e-4	1.32	0.64
1e-3	18,176	1.65e-5	2.16	0.89
1e-4	41,472	1.55e-6	2.87	1.03

**Fig. 1** Example 4.3: Discontinuous coefficient in 2D at $t = 0.1$. Adaptive sparse grid DG with $N = 8$ and $\epsilon = 10^{-4}$

with $r = (\sum_{i=1}^d x_i^2)^{1/2}$ being the radius.

For small time t (before the wave front touch the outside boundary), the exact solution in 2D can be represented by an integral which is derived by Hadamard's method of descent:

$$u(x_1, x_2, t) = \frac{1}{2\pi} \iint_{\rho < t} \frac{100e^{-500(y_1^2 + y_2^2)}}{\sqrt{t^2 - \rho^2}} dy_1 dy_2 \quad (4.12)$$

with $\rho := \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2}$ and then computed by using numerical integrations with sufficiently small error tolerance. For $d = 3$, there exists the analytic solution:

$$u(\mathbf{x}, t) = \frac{1}{20r} \left(e^{-500(t-r)^2} - e^{-500(t+r)^2} \right). \quad (4.13)$$

The numerical results for 2D and 3D are presented in Figs. 2 and 3. In both cases, our numerical solutions coincide with the exact solutions quite well. The L^∞ errors between the numerical and the exact solutions at $t = 0.5$ are 9.06×10^{-5} and 6.79×10^{-4} for 2D and 3D, which are both in the same magnitude as the adaptive parameter $\epsilon = 1 \times 10^{-4}$. This indicates

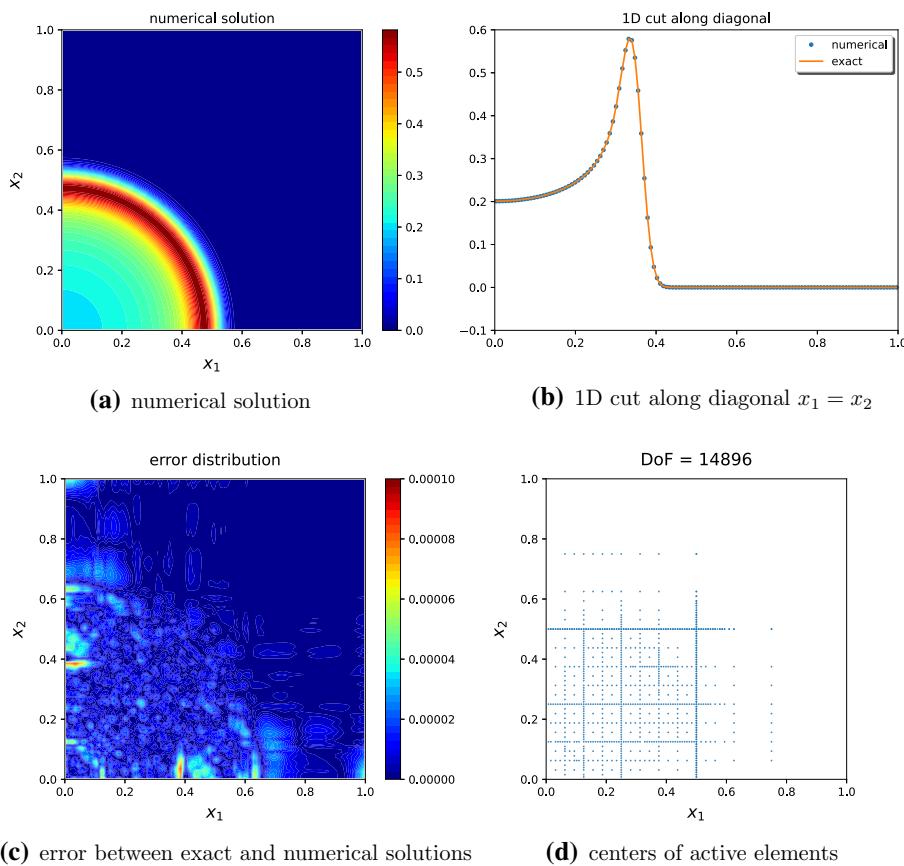


Fig. 2 Example 4.4: Expanding wave in homogeneous medium in 2D at $t = 0.5$. Adaptive sparse grid DG. $N = 7$ and $\epsilon = 10^{-4}$

that our adaptive algorithm controls the error really well. The DoFs are 14896 and 188672 for 2D and 3D. It can be also observed that the active elements in 3D are more “sparse” than 2D. This is a numerical evidence that the Huyghens principle only holds for wave equations in odd dimensions.

Example 4.5 (Isotropic wave propagation in heterogeneous media) We consider the wave equation with discontinuous coefficient on the computational domain $\Omega = [0, 1]^d$ for $d = 2$ and 3 [13].

$$c^2 = \begin{cases} \frac{1}{4}, & \text{if } 0.35 \leq x_1 \leq 0.65, \\ 1, & \text{otherwise.} \end{cases} \quad (4.14)$$

Note that the jump in material coefficient is not aligned with the cell interface on Ω_N .

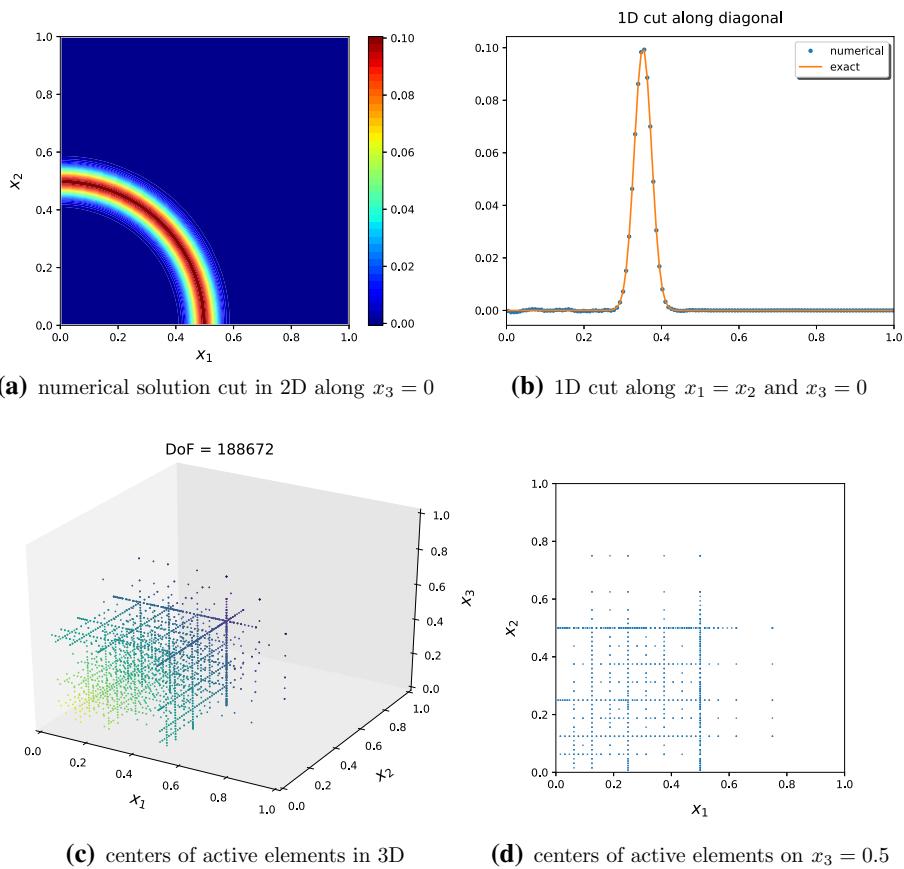


Fig. 3 Example 4.4: Expanding wave in homogeneous medium in 3D at $t = 0.5$. Adaptive sparse grid DG. $N = 7$ and $\epsilon = 10^{-4}$

For both 2D and 3D case, the initial conditions are taken as

$$u(\mathbf{x}, 0) = 0, \quad u_t(\mathbf{x}, 0) = 100e^{-500r^2}. \quad (4.15)$$

with $r = \left(\sum_{i=1}^d (x_i - \frac{1}{2})^2\right)^{\frac{1}{2}}$. The zero Dirichlet boundary conditions are used.

The profiles and centers of active elements obtained by the adaptive scheme are shown in Fig. 4 for 2D and Fig. 5 for 3D. We see that the wave fronts propagate at different speeds in these two media and our adaptive scheme capture this phenomenon and obtain comparable results to the literature [13].

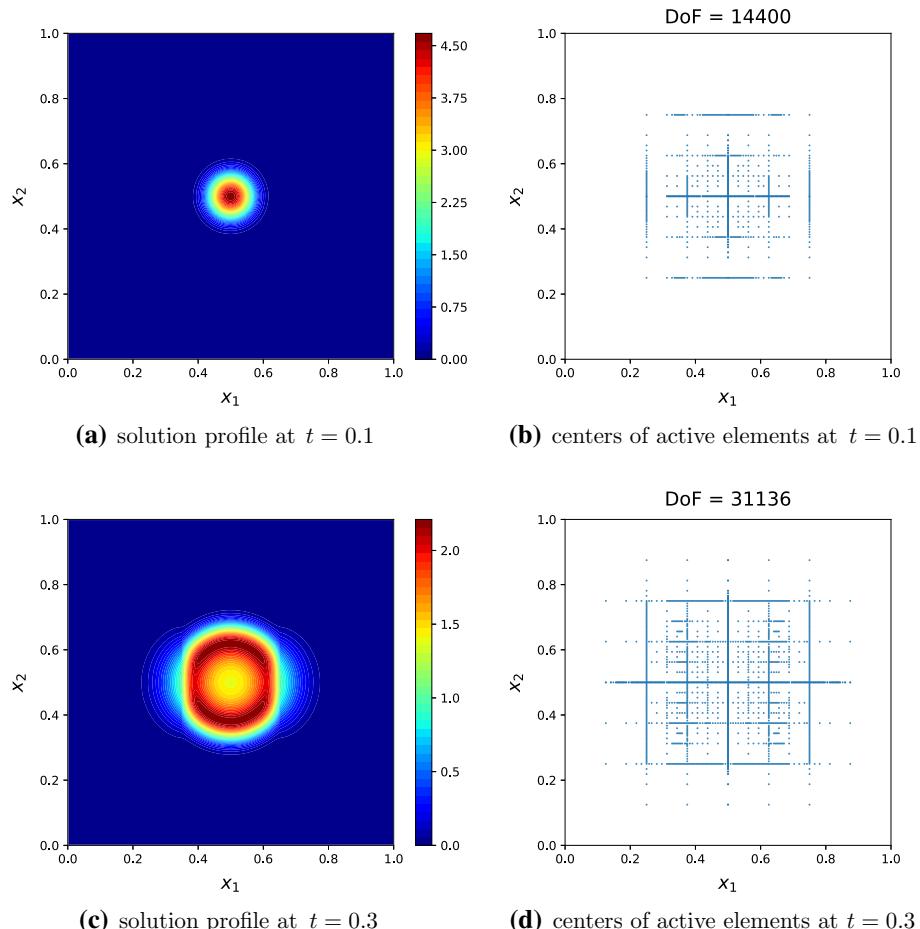


Fig. 4 Example 4.5: Isotropic wave propagation within heterogeneous media in 2D at $t = 0.1$ and $t = 0.3$. Adaptive sparse grid DG. $N = 8$ and $\epsilon = 10^{-4}$. Left: solution profile; right: centers of active elements

5 Conclusion

In this paper, we develop an adaptive multiresolution DG scheme for wave equations in second order form in multi-dimensions. Our method can achieve similar computational complexity as the sparse grid DG method for smooth solutions like those proposed for equations in first order form [21,22]. Extensive numerical tests in 2D and 3D verify the accuracy and robustness of the adaptive schemes for smooth and piecewise smooth wave propagation speed. Though the formulation is based on IPDG scheme for scalar wave equation, it can be extended to other DG method and other wave applications. Future work includes extensions to other boundary conditions and investigation on stability of schemes with interpolation. In an effort for promoting reproducible research, the code generating the results in this paper can be found at the github link: <https://github.com/JuntaoHuang/adaptive-multiresolution-DG>.

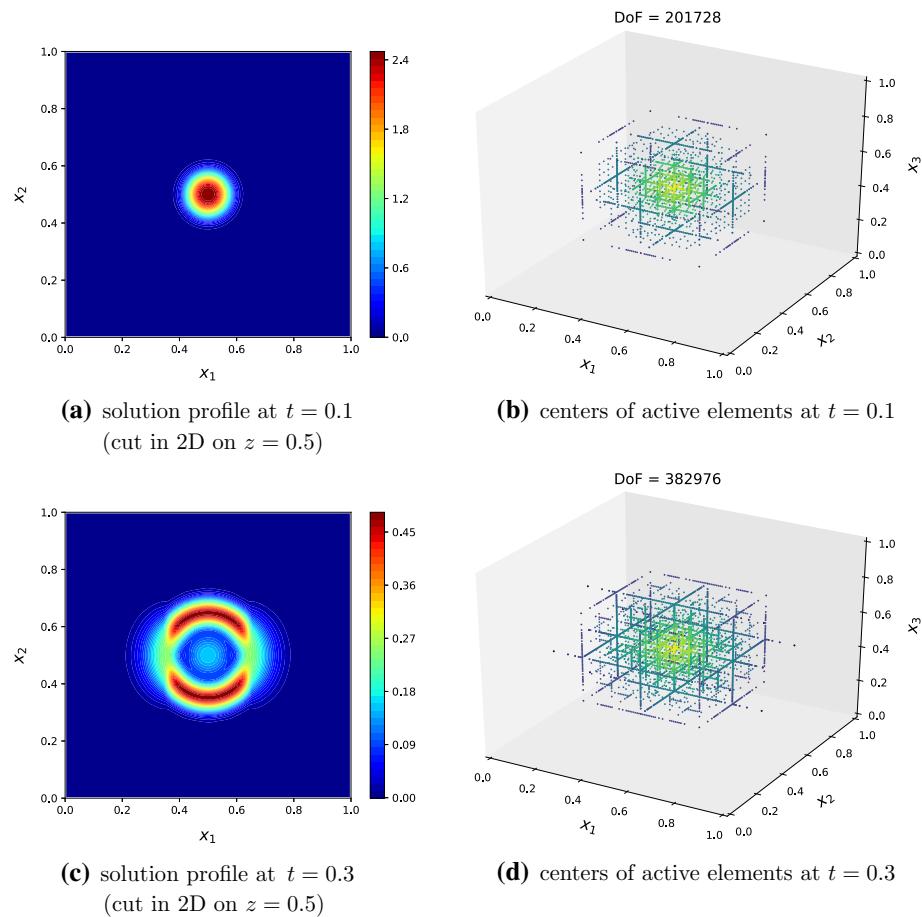


Fig. 5 Example 4.5: Isotropic wave propagation within heterogeneous media in 3D at $t = 0.1$ and $t = 0.3$. Adaptive sparse grid DG. $N = 7$ and $\epsilon = 10^{-4}$. Left: solution profile; right: centers of active elements

Acknowledgements We would like to thank Daniel Appelö for discussions on numerical examples of wave propagation, Qi Tang and Kai Huang for the assistance and discussion in code implementation.

Interpolation Basis Functions

For completeness of the paper, we present details of the multiresolution interpolation basis functions, which are first introduced in [37]. We will first focus on the case in which the interpolation points are imposed in the inner domain, as implemented in Table 5. Then we discuss the case in which the points includes the cell interface points. Here, we only discuss the case when $M = 4$ and $M = 5$. For $M = 1, 2, 3$, we refer readers to the appendix in [27].

The basis functions in \tilde{W}_1 are piecewise polynomials on $I_l := (0, \frac{1}{2})$ and $I_r := (\frac{1}{2}, 1)$. Note that the functions may be discontinuous at the interface $x = 1/2$, thus I_l and I_r are both defined to be open intervals. The basis functions in \tilde{W}_1 in this paper are all supported on one half interval I_l or I_r and vanish on the other half. For simplicity, we will only declare the

function on its support. For example, $\psi_0(x)|_{I_r}$ gives the definition of ψ_0 on I_r and indicates that ψ_0 vanishes on I_l .

Interpolation Points in the Inner Domain

$M = 4$

The interpolation points are

$$\tilde{X}_0 = \left\{ \frac{1}{6}, \frac{7}{24}, \frac{1}{3}, \frac{7}{12}, \frac{2}{3} \right\}, \quad \tilde{X}_1 = \left\{ \frac{1}{12}, \frac{7}{48}, \frac{31}{48}, \frac{19}{24}, \frac{5}{6} \right\}.$$

The basis functions in \tilde{W}_0^4 and \tilde{W}_1^4 are

$$\begin{aligned} \phi_0(x) &= \frac{4}{45}(3x-2)(3x-1)(12x-7)(24x-7), \\ \phi_1(x) &= -\frac{512}{189}(3x-2)(3x-1)(6x-1)(12x-7), \\ \phi_2(x) &= \frac{1}{3}(3x-2)(6x-1)(12x-7)(24x-7), \\ \phi_3(x) &= -\frac{32}{105}(3x-2)(3x-1)(6x-1)(24x-7), \\ \phi_4(x) &= \frac{1}{27}(3x-1)(6x-1)(12x-7)(24x-7). \end{aligned}$$

and

$$\begin{aligned} \psi_0(x)|_{I_l} &= \frac{8}{45}(3x-1)(6x-1)(24x-7)(48x-7), \\ \psi_1(x)|_{I_l} &= -\frac{1024}{189}(3x-1)(6x-1)(12x-1)(24x-7), \\ \psi_2(x)|_{I_r} &= -\frac{1024}{189}(3x-2)(6x-5)(12x-7)(24x-19), \\ \psi_3(x)|_{I_r} &= -\frac{64}{105}(3x-2)(6x-5)(12x-7)(48x-31), \\ \psi_4(x)|_{I_r} &= \frac{2}{27}(3x-2)(12x-7)(24x-19)(48x-31) \end{aligned}$$

$M = 5$

The interpolation points are

$$\tilde{X}_0 = \left\{ \frac{1}{12}, \frac{1}{6}, \frac{7}{24}, \frac{1}{3}, \frac{7}{12}, \frac{2}{3} \right\}, \quad \tilde{X}_1 = \left\{ \frac{7}{48}, \frac{1}{24}, \frac{31}{48}, \frac{19}{24}, \frac{5}{6}, \frac{13}{24} \right\}.$$

The basis functions in \tilde{W}_0^5 and \tilde{W}_1^5 are

$$\begin{aligned} \phi_0(x) &= \frac{1}{315}(-16)(3x-2)(3x-1)(6x-1)(12x-7)(24x-7), \\ \phi_1(x) &= \frac{4}{45}(3x-2)(3x-1)(12x-7)(12x-1)(24x-7), \\ \phi_2(x) &= -\frac{1024}{945}(3x-2)(3x-1)(6x-1)(12x-7)(12x-1), \\ \phi_3(x) &= \frac{1}{9}(3x-2)(6x-1)(12x-7)(12x-1)(24x-7), \\ \phi_4(x) &= -\frac{16}{315}(3x-2)(3x-1)(6x-1)(12x-1)(24x-7), \\ \phi_5(x) &= \frac{1}{189}(3x-1)(6x-1)(12x-7)(12x-1)(24x-7), \end{aligned}$$

and

$$\begin{aligned}\psi_0(x)|_{I_l} &= -\frac{2048}{945}(3x-1)(6x-1)(12x-1)(24x-7)(24x-1), \\ \psi_1(x)|_{I_l} &= -\frac{32}{315}(3x-1)(6x-1)(12x-1)(24x-7)(48x-7), \\ \psi_2(x)|_{I_r} &= -\frac{2048}{945}(3x-2)(6x-5)(12x-7)(24x-19)(24x-13), \\ \psi_3(x)|_{I_r} &= -\frac{32}{315}(3x-2)(6x-5)(12x-7)(24x-13)(48x-31), \\ \psi_4(x)|_{I_r} &= \frac{2}{189}(3x-2)(12x-7)(24x-19)(24x-13)(48x-31), \\ \psi_5(x)|_{I_r} &= -\frac{32}{315}(3x-2)(6x-5)(12x-7)(24x-19)(48x-31)\end{aligned}$$

Interpolation Points with the Interface Points

M = 4

The interpolation points are

$$\tilde{X}_0 = \left\{ 0^+, \left(\frac{1}{4}\right)^-, \left(\frac{1}{2}\right)^-, \left(\frac{3}{4}\right)^-, 1^- \right\}, \quad \tilde{X}_1 = \left\{ \left(\frac{1}{8}\right)^-, \left(\frac{3}{8}\right)^-, \left(\frac{1}{2}\right)^+, \left(\frac{5}{8}\right)^-, \left(\frac{7}{8}\right)^- \right\}.$$

The basis functions in \tilde{W}_0^4 and \tilde{W}_1^4 are

$$\begin{aligned}\phi_0(x) &= \frac{1}{3}(x-1)(2x-1)(4x-3)(4x-1), \\ \phi_1(x) &= -\frac{16}{3}(x-1)x(2x-1)(4x-3), \\ \phi_2(x) &= 4(x-1)x(4x-3)(4x-1), \\ \phi_3(x) &= -\frac{16}{3}(x-1)x(2x-1)(4x-1), \\ \phi_4(x) &= \frac{1}{3}x(2x-1)(4x-3)(4x-1).\end{aligned}$$

and

$$\begin{aligned}\psi_0(x)|_{I_l} &= -\frac{32}{3}x(2x-1)(4x-1)(8x-3), \\ \psi_1(x)|_{I_l} &= -\frac{32}{3}x(2x-1)(4x-1)(8x-1), \\ \psi_2(x)|_{I_r} &= \frac{2}{3}(x-1)(4x-3)(8x-7)(8x-5), \\ \psi_3(x)|_{I_r} &= -\frac{32}{3}(x-1)(2x-1)(4x-3)(8x-7), \\ \psi_4(x)|_{I_r} &= -\frac{32}{3}(-32)(x-1)(2x-1)(4x-3)(8x-5)\end{aligned}$$

M = 5

The interpolation points are

$$\tilde{X}_0 = \left\{ 0^+, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1^- \right\}, \quad \tilde{X}_1 = \left\{ \frac{1}{10}, \frac{3}{10}, \left(\frac{1}{2}\right)^-, \left(\frac{1}{2}\right)^+, \frac{7}{10}, \frac{9}{10} \right\}.$$

The basis functions in \tilde{W}_0^5 and \tilde{W}_1^5 are

$$\begin{aligned}\phi_0(x) &= -\frac{1}{24}(x-1)(5x-4)(5x-3)(5x-2)(5x-1), \\ \phi_1(x) &= \frac{25}{24}(x-1)x(5x-4)(5x-3)(5x-2), \\ \phi_2(x) &= -\frac{25}{12}(x-1)x(5x-4)(5x-3)(5x-1), \\ \phi_3(x) &= \frac{25}{12}(x-1)x(5x-4)(5x-2)(5x-1), \\ \phi_4(x) &= -\frac{25}{24}(x-1)x(5x-3)(5x-2)(5x-1), \\ \phi_5(x) &= \frac{1}{24}x(5x-4)(5x-3)(5x-2)(5x-1),\end{aligned}$$

and

$$\begin{aligned}\psi_0(x)|_{I_l} &= \frac{25}{3}x(2x-1)(5x-2)(5x-1)(10x-3), \\ \psi_1(x)|_{I_l} &= \frac{50}{3}x(2x-1)(5x-2)(5x-1)(10x-1), \\ \psi_2(x)|_{I_r} &= \frac{1}{3}x(5x-2)(5x-1)(10x-3)(10x-1), \\ \psi_3(x)|_{I_r} &= -\frac{1}{3}(x-1)(5x-4)(5x-3)(10x-9)(10x-7), \\ \psi_4(x)|_{I_r} &= -\frac{50}{3}(x-1)(2x-1)(5x-4)(5x-3)(10x-9), \\ \psi_5(x)|_{I_r} &= -\frac{25}{3}(x-1)(2x-1)(5x-4)(5x-3)(10x-7).\end{aligned}$$

References

1. Ainsworth, M., Monk, P., Muniz, W.: Dispersive and dissipative properties of discontinuous Galerkin finite element methods for the second-order wave equation. *J. Sci. Comput.* **27**(1–3), 5–40 (2006)
2. Alpert, B.: A class of bases in L^2 for the sparse representation of integral operators. *SIAM J. Math. Anal.* **24**(1), 246–262 (1993)
3. Appelö, D., Hagstrom, T.: A new discontinuous Galerkin formulation for wave equations in second-order form. *SIAM J. Numer. Anal.* **53**(6), 2705–2726 (2015)
4. Arnold, D.N.: An interior penalty finite element method with discontinuous elements. *SIAM J. Numer. Anal.* **19**(4), 742–760 (1982)
5. Arnold, D.N., Brezzi, F., Cockburn, B., Marini, L.D.: Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.* **39**(5), 1749–1779 (2002)
6. Berger, M.J., Colella, P.: Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.* **82**(1), 64–84 (1989)
7. Berger, M.J., Oliger, J.: Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.* **53**(3), 484–512 (1984)
8. Bokanowski, O., Gärcke, J., Griebel, M., Klompmaker, I.: An adaptive sparse grid semi-Lagrangian scheme for first order Hamilton-Jacobi Bellman equations. *J. Sci. Comput.* **55**(3), 575–605 (2013)
9. Brown, D.L., Cheshire, G.S., Henshaw, W.D., Quinlan, D.J.: Overture: an object-oriented software system for solving partial differential equations in serial and parallel environments. Technical report, Los Alamos National Lab, NM (United States) (1997)
10. Bungartz, H.-J., Griebel, M.: Sparse grids. *Acta Numerica* **13**, 147–269 (2004)
11. Burstedde, C., Wilcox, L.C., Ghattas, O.: p4est: scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM J. Sci. Comput.* **33**(3), 1103–1133 (2011)
12. Calle, J.L.D., Devloo, P.R.B., Gomes, S.M.: Wavelets and adaptive grids for the discontinuous Galerkin method. *Numer. Algorithms* **39**(1–3), 143–154 (2005)
13. Chou, C.-S., Shu, C.-W., Xing, Y.: Optimal energy conserving local discontinuous Galerkin methods for second-order wave equation in heterogeneous media. *J. Comput. Phys.* **272**, 88–107 (2014)
14. Cockburn, B., Hou, S., Shu, C.-W.: The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case. *Math. Comput.* **54**(190), 545–581 (1990)

15. Cockburn, B., Karniadakis, G., Shu, C.-W.: The development of discontinuous Galerkin methods. In: Cockburn, B., Karniadakis, G., Shu, C.-W. (eds.) *Discontinuous Galerkin Methods: Theory, Computation and Applications*, vol. 11, pp. 3–50. Springer, Berlin (2000)
16. Cohen, G.C.: *Higher-Order Numerical Methods for Transient Wave Equations*. Springer, Berlin (2002)
17. Etienne, V., Chaljub, E., Virieux, J., Glinsky, N.: An hp-adaptive discontinuous Galerkin finite-element method for 3-d elastic wave modelling. *Geophys. J. Int.* **183**(2), 941–962 (2010)
18. Gottlieb, D., Orszag, S.A.: *Numerical Analysis Of Spectral Methods: Theory and Applications*. SIAM, Philadelphia (1977)
19. Gottlieb, S., Shu, C.-W., Tadmor, E.: Strong stability preserving high order time discretization methods. *SIAM Rev.* **43**, 89–112 (2001)
20. Grote, M.J., Schneebeli, A., Schötzau, D.: Discontinuous Galerkin finite element method for the wave equation. *SIAM J. Numer. Anal.* **44**(6), 2408–2431 (2006)
21. Guo, W., Cheng, Y.: A sparse grid discontinuous Galerkin method for high-dimensional transport equations and its application to kinetic simulations. *SIAM J. Sci. Comput.* **38**(6), A3381–A3409 (2016)
22. Guo, W., Cheng, Y.: An adaptive multiresolution discontinuous Galerkin method for time-dependent transport equations in multidimensions. *SIAM J. Sci. Comput.* **39**(6), A2962–A2992 (2017)
23. Gustafsson, B., Kreiss, H.-O., Oliger, J.: *Time Dependent Problems and Difference Methods*, vol. 24. Wiley, Hoboken (1995)
24. Henshaw, W.D.: A high-order accurate parallel solver for Maxwell's equations on overlapping grids. *SIAM J. Sci. Comput.* **28**(5), 1730–1765 (2006)
25. Hesthaven, J., Warburton, T.: Nodal high-order methods on unstructured grids. I time-domain solution of Maxwell's equations. *J. Comput. Phys.* **181**, 186–221 (2002)
26. Hovhannисyan, N., Müller, S., Schäfer, R.: Adaptive multiresolution discontinuous Galerkin schemes for conservation laws. *Math. Comput.* **83**(285), 113–151 (2014)
27. Huang, J., and Cheng, Y.: An adaptive multiresolution discontinuous Galerkin method with artificial viscosity for scalar hyperbolic conservation laws in multidimensions (2019). arXiv preprint [arXiv:1906.00829](https://arxiv.org/abs/1906.00829)
28. Huang, J., Shu, C.-W.: Error estimates to smooth solutions of semi-discrete discontinuous Galerkin methods with quadrature rules for scalar conservation laws. *Numer. Methods Partial Differ. Equ.* **33**(2), 467–488 (2017)
29. Joly, P.: *Variational Methods for Time-Dependent Wave Propagation Problems*. Volume 31 of *Topics in Computational Wave Propagation*, pp. 201–264. Springer, Berlin (2003)
30. Käser, M., Dumbser, M.: An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes. the two-dimensional isotropic case with external source terms. *Geophys. J. Int.* **166**(2), 855–877 (2006)
31. Mallat, S.: *A Wavelet Tour of Signal Processing*. Elsevier, Amsterdam (1999)
32. Reed, W., Hill, T.: Tiangular mesh methods for the neutron transport equation. Technical report, Los Alamos National Laboratory, Los Alamos, NM (1973)
33. Seriani, G., Priolo, E.: Spectral element method for acoustic wave simulation in heterogeneous media. *Finite Elem. Anal. Des.* **16**(3–4), 337–348 (1994)
34. Shen, J., Yu, H.: Efficient spectral sparse grid methods and applications to high-dimensional elliptic problems. *SIAM J. Sci. Comput.* **32**(6), 3228–3250 (2010)
35. Shu, C.-W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.* **77**(2), 439–471 (1988)
36. Sjögreen, B., Petersson, N.A.: A fourth order accurate finite difference scheme for the elastic wave equation in second order formulation. *J. Sci. Comput.* **52**(1), 17–48 (2012)
37. Tao, Z., Jiang, Y., Cheng, Y.: An adaptive high-order piecewise polynomial based sparse grid collocation method with applications (2019). arXiv preprint [arXiv:1912.03982](https://arxiv.org/abs/1912.03982)
38. Tromp, J., Komatitsch, D., Liu, Q.: Spectral-element and adjoint methods in seismology. *Commun. Comput. Phys.* **3**(1), 1–32 (2008)
39. Wang, Z., Tang, Q., Guo, W., Cheng, Y.: Sparse grid discontinuous Galerkin methods for high-dimensional elliptic equations. *J. Comput. Phys.* **314**, 244–263 (2016)
40. Wilcox, L.C., Stadler, G., Burstedde, C., Ghattas, O.: A high-order discontinuous Galerkin method for wave propagation through coupled elastic-acoustic media. *J. Comput. Phys.* **229**(24), 9373–9396 (2010)
41. Xing, Y., Chou, C.-S., Shu, C.-W.: Energy conserving local discontinuous Galerkin methods for wave propagation problems. *Inverse Probl. Imaging* **7**(3), 967 (2013)
42. Zeiser, A.: Fast matrix-vector multiplication in the sparse-grid Galerkin method. *J. Sci. Comput.* **47**(3), 328–346 (2011)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Juntao Huang¹ · Yuan Liu² · Wei Guo³ · Zhanjing Tao⁴ · Yingda Cheng⁵

Juntao Huang
huangj75@msu.edu

Yuan Liu
liu@math.wichita.edu

Wei Guo
weimath.guo@ttu.edu

Yingda Cheng
ycheng@msu.edu

¹ Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA

² Department of Mathematics, Statistics and Physics, Wichita State University, Wichita, KS 67260, USA

³ Department of Mathematics and Statistics, Texas Tech University, Lubbock, TX 79409, USA

⁴ School of Mathematics, Jilin University, Changchun 130012, Jilin, China

⁵ Department of Mathematics, Department of Computational Mathematics, Science and Engineering, Michigan State University, East Lansing, MI 48824, USA