EFFICIENT QUADRATURE RULES FOR COMPUTING THE STIFFNESS MATRICES OF MASS-LUMPED TETRAHEDRAL ELEMENTS FOR LINEAR WAVE PROBLEMS[*]

S. GEEVERS[†], W. A. MULDER[‡], AND J. J. W. VAN DER VEGT[†]

Abstract. We present new and efficient quadrature rules for computing the stiffness matrices of mass-lumped tetrahedral elements for wave propagation modeling. These quadrature rules allow for a more efficient implementation of the mass-lumped finite element method and can handle materials that are heterogeneous within the element without loss of the convergence rate. The quadrature rules are designed for the specific function spaces of recently developed mass-lumped tetrahedra, which consist of standard polynomial function spaces enriched with higher-degree bubble functions. For the degree-2 mass-lumped tetrahedron, the most efficient quadrature rule seems to be an existing 14-point quadrature rule, but for tetrahedra of degrees 3 and 4, we construct new quadrature rules that require fewer integration points than those currently available in the literature. Several numerical examples confirm that this approach is more efficient than computing the stiffness matrix exactly and that an optimal order of convergence is maintained, even when material properties vary within the element.

Key words. mass lumping, tetrahedral element, spectral element method, wave equation, quadrature rule

AMS subject classifications. 65M12, 65M60

DOI. 10.1137/18M1198557

1. Introduction. Mass-lumped tetrahedral element methods are efficient methods for solving linear wave equations, such as the acoustic wave equation, the elastic wave equations, or Maxwell's equations, on complex three-dimensional (3D) domains with sharp material interfaces [25]. They offer the same convergence rate and geometric flexibility as standard continuous tetrahedral element methods, but also allow for explicit time-stepping due to a diagonal mass matrix.

To obtain mass-lumped elements, Lagrangian basis functions are combined with an inexact quadrature rule for computing the mass matrix. A lumped matrix is obtained when the quadrature points coincide with the basis function nodes. For quadrilateral and hexahedral elements, mass-lumping is achieved using tensor-product basis functions and Gauss–Lobatto points, resulting in the well-known spectral element method [20, 21, 15]. For linear triangular and tetrahedral elements, mass-lumping is achieved with standard Lagrangian basis functions and a Newton–Cotes integration rule. For higher-degree triangular and tetrahedral elements, however, the element space needs to be enriched with higher-degree bubble functions in order to maintain stability and an optimal order of convergence [6]. So far, mass-lumped triangular elements of degrees 2 and 3 [5, 7], 4 [17], 5 [3], 6 [18], and 7–9 [16, 9] have been found. The first higher-degree tetrahedral elements were presented in [17] for degree 2 and

[†]Department of Applied Mathematics, University of Twente, Enschede, The Netherlands (s.geevers@utwente.nl, j.j.w.vandervegt@utwente.nl).

[‡]Shell Global Solutions International BV, Amsterdam, The Netherlands, and Delft University of Technology, Delft, The Netherlands (wim.mulder@shell.com).

in [3] for degree 3. Recently, we presented new mass-lumped tetrahedral elements of degrees 2 to 4 in [12]. The new degree-2 and degree-3 elements require significantly fewer nodes than the earlier versions, while mass-lumped tetrahedra of degree 4 had not been found before. Because of the reduced number of nodes, these new mass-lumped elements are also much more efficient than the earlier versions [12] and are therefore more suitable for large-scale 3D simulations.

A question that remains is how to efficiently compute the stiffness matrix for these elements. When the material parameters are piecewise constant, the stiffness matrix can be evaluated exactly [19]. Alternatively, we can use a quadrature rule to approximate the stiffness matrix. This latter approach can significantly reduce the number of computations, as we will demonstrate in section 6. Moreover, it also allows us to handle material parameters that vary within the element without loss of convergence rate, as we will prove in section 4.

Finding an efficient quadrature rule for the stiffness matrix for mass-lumped tetrahedra is not straightforward. For hexahedral elements, the stiffness matrix can be approximated with the same Gauss–Lobatto quadrature rule that is used for the mass matrix, but for mass-lumped tetrahedra, using the quadrature rule of the mass matrix to also evaluate the stiffness matrix turns out to be inefficient or inaccurate. In this paper, we therefore present new and efficient quadrature rules for computing the stiffness matrices for mass-lumped tetrahedra.

To obtain these quadrature rules, we show that the quadrature rule only needs to be exact for functions in the space $\mathcal{P}_{p-1} \otimes D\tilde{U}$, where $p \geq 2$ denotes the degree of the element, $\mathcal{P}_{p-1}$ denotes the space of polynomials up to degree $p-1$, and $D\tilde{U}$ denotes the space of partial derivatives of functions in the element space. Since the mass-lumped tetrahedra contain higher-degree bubble functions, so does the space $\mathcal{P}_{p-1} \otimes D\tilde{U}$ that needs to be integrated exactly. Most quadrature rules in literature, however, are designed to be exact for spaces of the form $\mathcal{P}_k$. Such quadrature rules for tetrahedral domains can be found in, for example, [22, 13, 14, 8, 24, 23] and the references therein. We could choose k equal to the highest polynomial degree that appears in $\mathcal{P}_{p-1} \otimes D\tilde{U}$, but the resulting number of quadrature points may then be suboptimal. Instead, we try to find quadrature rules that are exact for $\mathcal{P}_{p-1} \otimes D\tilde{U}$ with a minimal number of quadrature points.

For the degree-2 tetrahedral element, the most efficient quadrature rule still seems to be the 14-point rule of [13] that is accurate for polynomials up to degree 5. For the degree-3 element and the three degree-4 elements of 60, 61, and 65 nodes, however, we present new quadrature rules that require 21, 51, 60, and 60 points, respectively, while using a quadrature rule from the current literature would require 24 [14], 59 [24], 79 [24], and 79 [24] points.

This paper is organized as follows. In section 2, we introduce the mass-lumped finite element method, and in section 3, we present our new quadrature rules for evaluating the stiffness matrix. We prove in section 4 that the conditions used to obtain our quadrature rules result in optimal convergence rates. In section 5, we analyze and compare the dispersion properties and resulting time step size for our new quadrature rules and several other rules available in the literature. In section 6, we show numerical examples demonstrating that using our numerical quadrature rules for evaluating the stiffness matrix is more efficient than evaluating the integrals exactly and that the convergence rate is not lost when material parameters vary within the elements. Finally, we summarize our main conclusions in section 7.

2. The mass-lumped finite element method. To present and analyze the mass-lumped finite element method, we consider the scalar wave equation given by

$$\rho \partial_t^2 u = \nabla \cdot c \nabla u + f \qquad \text{in } \Omega \times (0, T), \tag{1a}$$

$$u = 0 \qquad \text{on } \partial\Omega \times (0, T), \tag{1b}$$

$$u|_{t=0} = u_0 \qquad \text{in } \Omega, \tag{1c}$$

$$\partial_t u|_{t=0} = v_0 \qquad \text{in } \Omega, \tag{1d}$$

where $\Omega \subset \mathbb{R}^3$ is the spatial domain, $(0, T)$ is the time domain, $u : \Omega \times (0, T) \to \mathbb{R}$ is the scalar field that needs to be solved, ∇ is the gradient operator, $f : \Omega \times (0, T) \to \mathbb{R}$ is the source term, $u_0, v_0 : \Omega \to \mathbb{R}$ are the initial values, and $\rho, c : \Omega \to \mathbb{R}^+$ are positive spatial parameters. The spatial domain Ω is assumed to be a bounded open domain with Lipschitz boundary $\partial\Omega$, and the parameters ρ and c are assumed to be bounded by $\rho_0 \le \rho \le \rho_1$ and $c_0 \le c \le c_1$, with ρ_0, ρ_1, c_0, c_1 strictly positive constants.

To solve the scalar wave equation with a finite element method, we consider the weak formulation. Let $L^2(\Omega)$ denote the standard Lesbesque space of square-integrable functions on Ω, $H_0^1(\Omega)$ the standard Sobolev space of functions in $L^2(\Omega)$ that vanish on $\partial\Omega$ and have square-integrable weak derivatives, and $L^2(0, T; U)$, with U a Banach space, the Bochner space of functions $f : (0, T) \to U$ such that $\|f\|_U$ is square integrable on $(0, T)$. Assume $u_0 \in H_0^1(\Omega)$, $v_0 \in L^2(\Omega)$, and $f \in L^2(0, T; L^2(\Omega))$. The weak formulation of (1) can then be written as finding $u \in L^2(0, T; H_0^1(\Omega))$, with $\partial_t u \in L^2(0, T; L^2(\Omega))$ and $\partial_t(\rho \partial_t u) \in L^2(0, T; H^{-1}(\Omega))$, such that $u|_{t=0} = u_0$, $\partial_t u|_{t=0} = v_0$, and

$$\langle \partial_t(\rho \partial_t u), w \rangle + (c \nabla u, \nabla w) = (f, w) \qquad \text{for all } w \in H_0^1(\Omega), \text{ a.e. } t \in (0, T), \tag{2}$$

where $(\cdot, \cdot)$ denotes the standard $L^2(\Omega)$ inner-product and $\langle \cdot, \cdot \rangle$ denotes the pairing between $H^{-1}(\Omega)$ and $H_0^1(\Omega)$.

This weak form of the wave equation can be solved with the mass-lumped finite element method, which consists of the following components:

- a tetrahedral mesh $\mathcal{T}_h$, where h denotes the radius of the smallest sphere that can contain each element,
- a reference tetrahedron $\tilde{e}$ with reference space $\tilde{U} = \mathcal{P}_p \oplus \tilde{U}^+ := \{u \mid u = w + u^+ \text{ for some } w \in \mathcal{P}_p, u^+ \in \tilde{U}^+\}$, where $\mathcal{P}_p$ denotes the space of polynomials of degree p or less and $\tilde{U}^+$ a space of higher-degree face and interior bubble functions,
- a set of reference nodes $\tilde{\mathcal{Q}}$ that can be used for both interpolation and quadrature on $\tilde{e}$,
- a set of quadrature weights $\{\tilde{\omega}_{\tilde{\mathbf{x}}}\}_{\tilde{\mathbf{x}} \in \tilde{\mathcal{Q}}}$.

Using these components, a finite element space can be constructed of the form

$$U_h = H_0^1(\Omega) \cap U(\mathcal{T}_h, \tilde{U}),$$

where

$$U(\mathcal{T}_h, \tilde{U}) := \{u \in H^1(\Omega) \mid u \circ \phi_e \in \tilde{U} \text{ for all } e \in \mathcal{T}_h\},$$

with $\phi_e : \tilde{e} \to e$ the reference-to-physical element mapping. The interpolation points are given by $\mathcal{Q}_h = \mathcal{Q}(\mathcal{T}_h, \tilde{\mathcal{Q}})$, where

$$\mathcal{Q}(\mathcal{T}_h, \tilde{\mathcal{Q}}) := \bigcup_{e \in \mathcal{T}_h} \bigcup_{\tilde{\mathbf{x}} \in \tilde{\mathcal{Q}}} \phi_e(\tilde{\mathbf{x}}),$$

and the $L^2(\Omega)$ inner-product is approximated by

$$(u, w) = \sum_{e \in \mathcal{T}_h} \frac{|e|}{|\tilde{e}|} \int_{\tilde{e}} \tilde{u}_e \tilde{w}_e \, d\tilde{x} \approx \sum_{e \in \mathcal{T}_h} \sum_{\tilde{\mathbf{x}} \in \tilde{\mathcal{Q}}} \frac{|e|}{|\tilde{e}|} \omega_{\tilde{\mathbf{x}}} \tilde{u}_e(\tilde{\mathbf{x}}) \tilde{w}_e(\tilde{\mathbf{x}}) =: (u, w)_{\mathcal{Q}_h}$$

with $|e|$ and $|\tilde{e}|$ the volume of e and $\tilde{e}$, respectively, and $\tilde{u}_e := u \circ \phi_e$, $\tilde{w}_e := w \circ \phi_e$.

Assume $u_0, v_0, \rho, c \in \mathcal{C}^0(\overline{\Omega})$ and $f : \mathcal{C}^0(\overline{\Omega} \times [0, T])$ are all continuous. The finite element method can then be formulated as finding $u_h : [0, T] \to U_h$ such that $u_h|_{t=0} = I_h u_0$, $\partial_t u_h|_{t=0} = I_h v_0$, and

$$(3) \qquad (\rho \partial_t^2 u_h, w)_{\mathcal{Q}_h} + (c \nabla u_h, \nabla w) = (f, w)_{\mathcal{Q}_h} \qquad \text{for all } w \in U_h, t \in [0, T],$$

where I_h is the interpolation operator that interpolates a continuous function at the points $\mathcal{Q}_h$ by a function in $U(\mathcal{T}_h, \tilde{U})$.

Now let $\{\mathbf{x}_i\}_{i=1}^N$ be the set of all interpolation points $\mathcal{Q}_h$ that do not lie on the boundary $\partial\Omega$, and define nodal basis functions $\{w_i\}_{i=1}^N$ such that $w_i(\mathbf{x}_j) = \delta_{ij}$ for all $i, j = 1, \ldots, N$, with δ the Kronecker delta. Also define, for any continuous function $u \in \mathcal{C}(\overline{\Omega})$, the interpolation vector $\underline{u} \in \mathbb{R}^N$ such that $\underline{u}_i := u(\mathbf{x}_i)$ for all $i = 1, \ldots, N$. The finite element method can then be formulated as finding $\underline{u}_h : [0, T] \to \mathbb{R}^N$ such that $\underline{u}_h|_{t=0} = \underline{u_0}$, $\partial_t \underline{u}_h|_{t=0} = \underline{v_0}$, and

$$(4) \qquad \partial_t^2 \underline{u}_h + M^{-1} A \underline{u}_h = \underline{\rho^{-1} f} \qquad \text{for all } t \in [0, T].$$

Here, $M \in \mathbb{R}^{N \times N}$, with $M_{ij} := (\rho w_i, w_j)_{\mathcal{Q}_h}$, is the mass matrix, and $A \in \mathbb{R}^{N \times N}$, with $A_{ij} := (c \nabla w_i, \nabla w_j)$, is the stiffness matrix.

Since the interpolation points and quadrature points coincide, the mass matrix is diagonal with entries $M_{ii} = (\rho w_i, 1)_{\mathcal{Q}_h}$. Therefore, we can efficiently solve the system of ODEs in (4) using an explicit time-stepping scheme. Standard conforming finite element methods do not result in a (block-)diagonal mass matrix and are therefore less suitable for solving wave equations on large 3D meshes.

To remain accurate and stable, the mass-lumped finite element method needs to satisfy the following conditions [12]:

C1 (Unisolvent). The space $\tilde{U}$ is unisolvent on the nodes $\tilde{\mathcal{Q}}$.

C2 (Symmetry). The space $\tilde{U}$ and the set $\tilde{\mathcal{Q}}$ are invariant to affine mappings that map $\tilde{e}$ onto itself.

C3 (Face-conforming). If $\tilde{u} \in \tilde{U}$ is zero at all nodes in $\tilde{\mathcal{Q}} \cap \tilde{f}$, with $\tilde{f}$ a reference face, then $\tilde{u}$ is zero on $\tilde{f}$.

C4 (Positivity). The weights $\{\tilde{\omega}_{\tilde{\mathbf{x}}}\}_{\tilde{\mathbf{x}} \in \tilde{\mathcal{Q}}}$ are all strictly positive.

C5 (Accuracy). The quadrature rule is exact for functions in $\mathcal{P}_{p-2} \otimes \tilde{U}$ when $p \geq 2$.

The first three conditions are necessary to guarantee that the global basis functions are well-defined and continuous. The last two conditions are necessary for stability and for maintaining an optimal order of convergence.

When $p \geq 2$, these conditions cannot all be met for standard polynomial spaces $\tilde{U} = \mathcal{P}_p$. Therefore, the element space needs to be enriched with higher-degree bubble functions. We will focus on the mass-lumped tetrahedral elements recently presented in [12]. An overview of these elements is given in Table 1. There, n denotes the dimension of $\tilde{U}$, which is equal to the number of nodes per element, $B_f := \text{span}\{x_1 x_2 x_3, x_1 x_2 x_4, x_1 x_3 x_4, x_2 x_3 x_4\}$ denotes the span of the four face bubble functions, and $B_e := \text{span}\{x_1 x_2 x_3 x_4\}$ denotes the span of the element bubble function, with x_1, x_2, x_3, x_4 the four barycentric coordinates. We also used the notation

TABLE 1
Overview of mass-lumped tetrahedra.

p	n	$\tilde{U}$
2	15	$\mathcal{P}_2 \oplus B_f \oplus B_e$
3	32	$\mathcal{P}_3 \oplus B_f \mathcal{P}_1 \oplus B_e \mathcal{P}_1$
4	60	$\mathcal{P}_4 \oplus B_f \mathcal{P}_2 \oplus B_e(\mathcal{P}_2 + B_f)$
	61	$\mathcal{P}_4 \oplus B_f \mathcal{P}_2 \oplus B_e(\mathcal{P}_2 + B_f + B_e)$
	65	$\mathcal{P}_4 \oplus B_f(\mathcal{P}_2 + B_f) \oplus B_e(\mathcal{P}_2 + B_f + B_e)$

$UV := U \otimes V := \{w \mid w = uv \text{ for some } u \in U, v \in V\}$ for any two function spaces U, V.

To apply these elements more efficiently, we also approximate the L^2 inner-product for the stiffness matrix, $(c\nabla u, \nabla v)$, with a quadrature rule. This also allows us to handle material parameters c that vary within the element. It turns out that it is more efficient and sometimes even necessary to compute the stiffness matrix with a different quadrature rule than for the mass matrix. We will denote the quadrature points and weights for the stiffness matrix by $\tilde{\mathcal{Q}}'$ and $\{\tilde{\omega}'_{\tilde{\mathbf{x}}}\}_{\tilde{\mathbf{x}} \in \tilde{\mathcal{Q}}'}$, respectively, and denote the corresponding approximated $L^2(\Omega)$-product by $(\cdot, \cdot)_{\mathcal{Q}'_h}$.

The resulting finite element method remains stable and accurate if the following conditions are also satisfied:

C6 (Positivity). The weights $\{\tilde{\omega}'_{\tilde{\mathbf{x}}}\}_{\tilde{\mathbf{x}} \in \tilde{\mathcal{Q}}'}$ are all strictly positive.

C7 (Spurious-free). There is no function $\tilde{u} \in \tilde{U}$ with zero gradient $\tilde{\nabla}\tilde{u} = \mathbf{0}$ on all quadrature points $\tilde{\mathcal{Q}}'$ except the constant function. In case of linear elasticity, there is no function $\tilde{\mathbf{u}} \in \tilde{U}^3$ with zero strain $\tilde{\nabla}\tilde{\mathbf{u}} + \tilde{\nabla}\tilde{\mathbf{u}}^t = \mathbf{0}$ on all quadrature points $\tilde{\mathcal{Q}}'$ except the six rigid motions.

C8 (Accuracy). If $p \geq 2$, the quadrature rule for the stiffness matrix is exact for functions in $\mathcal{P}_{p-1} \otimes D\tilde{U}$, where $D\tilde{U}$ denotes the space of all partial derivatives of all functions in $\tilde{U}$.

A proof that these three conditions are sufficient to maintain an optimal order of convergence is given in section 4.

We constructed quadrature rules that satisfy these conditions for the specific function spaces of the higher-degree mass-lumped tetrahedra presented in Table 1. For the degree-2 element, the most efficient quadrature rule seems to be an existing 14-point rule that is fifth-order accurate, but for the higher-degree elements, we obtained new quadrature rules that require fewer points than those currently available in the literature. An overview of these rules is given in the next section.

3. Efficient quadrature rules for the stiffness matrix. To present the quadrature rules for the stiffness matrix, let $\tilde{e}$ be the reference tetrahedron with vertices at $(0, 0, 0)$, $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$. The barycentric coordinates of this element are given by the three Cartesian coordinates x_1, x_2, x_3 and the fourth coordinate $x_4 := 1 - x_1 - x_2 - x_3$. These coordinates are useful for describing $\mathcal{S}$, the space of affine mappings that map $\tilde{e}$ onto itself, since any function $s \in \mathcal{S}$ can be defined by a permutation of the barycentric coordinates. In particular, we can write $s(x_1, x_2, x_3) = (x_i, x_j, x_k)$ for some $i, j, k \in \{1, 2, 3, 4\}$, $i \neq j \neq k \neq i$, for any $s \in \mathcal{S}$.

Now, let $\{\mathbf{x}\}$ denote point $\mathbf{x}$ and all equivalent points $s(\mathbf{x})$, with $s \in \mathcal{S}$. The quadrature rule will consist of several equivalence classes $\{\mathbf{x}\}$ with quadrature weights that are the same within each equivalence class. To give an example of an equivalence class, consider the point (c_1, c_1, c_1). The barycentric coordinates of this point are given by $c_1, c_1, c_1, 1 - 3c_1$, so the equivalence class $\{(c_1, c_1, c_1)\}$ consists of the four

TABLE 2

Types of quadrature points. The third column shows the number of equivalent points for each type.

Type	Points	#	Description
[4]	$\{(1/4, 1/4, 1/4)\}$	1	center of tetrahedron
[3, 1]	$\{(c, c, c)\}$	4	on line through vertex and center
[2, 2]	$\{(d, d, 1/2 - d)\}$	6	on line through edge-midpoint and center
[2, 1, 1]	$\{(f_1, f_1, f_2)\}$	12	on plane through center and two vertices
[1, 1, 1, 1]	$\{(g_1, g_2, g_3)\}$	24	arbitrary position

points (c_1, c_1, c_1), $(1 - 3c_1, c_1, c_1)$, $(c_1, 1 - 3c_1, c_1)$, and $(c_1, c_1, 1 - 3c_1)$ when $c_1 \neq \frac{1}{4}$. An overview of the different types of points is given in Table 2. The configuration of a quadrature rule is given by the numbers K_4, K_{31}, K_{22}, K_{211}, and K_{1111}, which indicate that the quadrature rule has K_4 distinct points of type [4], K_{31} points of type [3, 1], K_{22} points of type [2, 2], etc.

To find a set of points and weights that satisfy accuracy condition C8, we construct a linear basis that spans $V \supset \mathcal{P}_{p-1} \otimes D\tilde{U}$. We describe this linear basis and linear span using the notation $\{f_1, f_2, \ldots, f_k\}$, which denotes the span of the functions $f_1, \ldots, f_k$ and all equivalent functions $f_i \circ s$, with $i = 1, \ldots, k$ and $s \in \mathcal{S}$. To give an example, all equivalent versions of $x_1^2 x_2^2 x_3 x_4$ are given by the six functions $x_1^2 x_2^2 x_3 x_4$, $x_1^2 x_2 x_3^2 x_4$, $x_1^2 x_2 x_3 x_4^2$, $x_1 x_2^2 x_3^2 x_4$, $x_1 x_2^2 x_3 x_4^2$, and $x_1 x_2 x_3^2 x_4^2$, so $\{x_1^2 x_2^2 x_3 x_4\}$ denotes the span of these six functions.

After having constructed a basis $\{f_1, f_2, \ldots, f_k\}$ for V, we search for a quadrature rule that has a configuration with k parameters. These parameters consist of location parameters and quadrature weights. Because of the symmetry, a quadrature rule that is exact for a function f is exact for all its equivalent functions. Therefore, to satisfy C8, we end up with a nonlinear system of k equations:

$$(5) \qquad \int_{\tilde{e}} f_i(\tilde{\mathbf{x}}) \, d\tilde{x} = \sum_{\tilde{\mathbf{x}} \in \tilde{\mathcal{Q}}'} \omega'_{\tilde{\mathbf{x}}} f_i(\tilde{\mathbf{x}}), \qquad\qquad i = 1, \ldots, k.$$

We obtain solutions of this system using Newton's method for a large number of different initial values and check for each solution if it satisfies C6 and C7. When we cannot find an admissible solution for configurations with k parameters, we increase k and try again. We continue this process until we find a suitable quadrature rule.

The complete algorithm for finding a quadrature rule of k parameters can be summarized as follows:

1. Construct basis functions $f_1, \ldots, f_k$, such that $\{f_1, f_2, \ldots, f_k\} \supset \mathcal{P}_{p-1} \otimes D\tilde{U}$.
2. Choose a configuration for the quadrature rule such that

$$\# \text{ parameters} = K_4 + 2K_{31} + 2K_{22} + 3K_{211} + 4K_{1111} = k.$$

3. Solve the system of equations given in (5) using Newton's method with a random initial guess.
4. If the method converges within a maximum number of iterations, check if the solution satisfies conditions C6 and C7.
5. Repeat steps 3 and 4 until a maximum number of trials is reached or an admissible solution is found.

The quadrature rules that were obtained in this way are given in Tables 3 to 6. There, # denotes the number of nodes in each equivalence class and $\beta_f := x_1 x_2 x_3$ and $\beta_e = x_1 x_2 x_3 x_4$ denote the face bubble function and interior bubble function, respectively.

TABLE 3

Quadrature rule of 14 points ($K_4 = 0$, $K_{31} = 2$, $K_{22} = 1$, $K_{211} = 0$, $K_{1111} = 0$) [13] for the stiffness matrix of the degree-2 15-node tetrahedron.

Nodes	#	ω'	Node parameters
$\{(c_1, c_1, c_1)\}$	4	0.01224884051939366	0.09273525031089123
$\{(c_2, c_2, c_2)\}$	4	0.01878132095300264	0.3108859192633006
$\{(d, d, \frac{1}{2} - d)\}$	6	0.007091003462846911	0.04550370412564965
$V = \mathcal{P}_5 = \{x_1, x_1^2 x_2, x_1^3 x_2^2, \beta_f x_1, \beta_f x_1 x_2, \beta_e x_1\}$			

TABLE 4

New quadrature rule of 21 points ($K_4 = 1$, $K_{31} = 2$, $K_{22} = 0$, $K_{211} = 1$, $K_{1111} = 0$) for the stiffness matrix of the degree-3 32-node tetrahedron.

Nodes	#	ω'	Node parameters
$\{(c_1, c_1, c_1)\}$	4	0.008382813462606309	0.08360982293995379
$\{(c_2, c_2, c_2)\}$	4	0.01062803097330636	0.3195556046935656
$\{(f_1, f_1, f_2)\}$	12	0.005973459577178217	$\begin{bmatrix} 0.06366100187501753 \\ 0.3362519222398494 \end{bmatrix}$
$\{(\frac{1}{4}, \frac{1}{4}, \frac{1}{4})\}$	1	0.01894177399687740	-
$V = \mathcal{P}_5 \oplus B_f \mathcal{P}_3$			
$= \{x_1, x_1^2 x_2, x_1^3 x_2^2, \beta_f x_1, \beta_f x_1^2 x_2, \beta_f^2, \beta_e x_1, \beta_e x_1 x_2\}$			

TABLE 5

New quadrature rule of 51 points ($K_4 = 1$, $K_{31} = 2$, $K_{22} = 1$, $K_{211} = 3$, $K_{1111} = 0$) for the stiffness matrix of the degree-4 60-node tetrahedron.

Nodes	#	ω'	Node parameters
$\{(c_1, c_1, c_1)\}$	4	0.001076330088382485	0.04010756377220036
$\{(c_2, c_2, c_2)\}$	4	0.006422430307819483	0.1881144601918900
$\{(d, d, \frac{1}{2} - d)\}$	6	0.003859721113202450	0.1124010568611476
$\{(f_{11}, f_{11}, f_{12})\}$	12	0.003162722714222902	$\begin{bmatrix} 0.04781990270450464 \\ 0.2053222493389064 \end{bmatrix}$
$\{(f_{21}, f_{21}, f_{22})\}$	12	0.004715130256124021	$\begin{bmatrix} 0.2347999378738287 \\ 0.03405863749492695 \end{bmatrix}$
$\{(f_{31}, f_{31}, f_{32})\}$	12	0.001320748780834370	$\begin{bmatrix} 0.4614535776221135 \\ 0.06693547308143162 \end{bmatrix}$
$\{(\frac{1}{4}, \frac{1}{4}, \frac{1}{4})\}$	1	0.003130077388468573	-
$V = \mathcal{P}_7 \oplus B_f(\mathcal{P}_5 \oplus B_f \mathcal{P}_3) \oplus B_e \mathcal{P}_5$			
$= \{x_1, x_1^2 x_2, x_1^3 x_2^2, x_1^4 x_2^3, \beta_f x_1, \beta_f x_1^2 x_2, \beta_f x_1^3 x_2^2, \beta_f^2 x_1, \beta_f^2 x_1^2 x_2, \beta_f^3, \dots$			
$\dots, \beta_e x_1, \beta_e x_1^2 x_2, \beta_e x_1^3 x_2^2, \beta_e \beta_f x_1, \beta_e \beta_f x_1 x_2, \beta_e^2 x_1\}$			

The quadrature rule with the least number of points we could find for the degree-2 15-node tetrahedron is the 14-point fifth-order accurate rule of [13]. We also found an accurate quadrature rule of 10 points with positive weights, but the resulting method was not accurate since condition C7 was not satisfied: it had one nonconstant mode with gradient equal to zero at all 10 points. We also considered the 15-point quadrature rule used for the mass matrix, which also satisfies C6–C8. However, this rule significantly increases the condition number of the element matrix and therefore results in a considerably smaller time step size, as shown in section 5.

The quadrature rules for the degree-3 and degree-4 elements are new and require fewer quadrature points than rules currently available in the literature, since most

TABLE 6

New quadrature rule of 60 *points* $(K_4 = 0, \ K_{31} = 3, \ K_{22} = 2, \ K_{211} = 3, \ K_{1111} = 0)$ *for the stiffness matrix of the degree-4 61- and 65-node tetrahedron.*

Nodes	#	ω'	Node parameters
$\{(c_1, c_1, c_1)\}$	4	0.001137453809249273	0.04091036488546224
$\{(c_2, c_2, c_2)\}$	4	0.006907244220995018	0.1942594527940223
$\{(c_3, c_3, c_3)\}$	4	0.004458749819772567	0.3166409312612929
$\{(d_1, d_1, \frac{1}{2} - d_1)\}$	6	0.001389883779363477	0.02776256108257648
$\{(d_2, d_2, \frac{1}{2} - d_2)\}$	6	0.004236295194116969	0.1022199785693040
$\{(f_{11}, f_{11}, f_{12})\}$	12	0.001788418107829456	$\begin{bmatrix} 0.03511432271187172 \\ 0.2097218125202450 \end{bmatrix}$
$\{(f_{21}, f_{21}, f_{22})\}$	12	0.003642034272731381	$\begin{bmatrix} 0.1790174868402900 \\ 0.03980830656880513 \end{bmatrix}$
$\{(f_{31}, f_{31}, f_{32})\}$	12	0.001477531071582210	$\begin{bmatrix} 0.4192720711456938 \\ 0.008950317872961031 \end{bmatrix}$

$$V = \mathcal{P}_8 \oplus B_f^2 \mathcal{P}_3 \oplus B_e(\mathcal{P}_5 \oplus B_f \mathcal{P}_3)$$
$$= \{x_1, x_1^2 x_2, x_1^3 x_2^2, x_1^4 x_2^3, x_1^4 x_2^4, \beta_f x_1, \beta_f x_1^2 x_2, \beta_f x_1^3 x_2^2, \beta_f^2 x_1, \beta_f^2 x_1^2 x_2, \beta_f^3, \dots$$
$$\dots, \beta_e x_1, \beta_e x_1^2 x_2, \beta_e x_1^3 x_2^2, \beta_e \beta_f x_1, \beta_e \beta_f x_1^2 x_2, \beta_e \beta_f^2, \beta_e^2 x_1, \beta_e^2 x_1 x_2\}$$

quadrature rules in the literature are constructed to be exact for a function space of the form $\mathcal{P}_k$ and not for the specific function spaces $\mathcal{P}_{p-1} \otimes D\tilde{U}$. To give an example, the highest polynomial degree of $D\tilde{U}$ of the degree-4 61- or 65-node tetrahedron is 7, so $\mathcal{P}_3 \otimes D\tilde{U}$ contains a polynomial of degree 10. A quadrature rule that is order-10 accurate already requires 79 quadrature points [24], while our quadrature rule for these elements only requires 60 points. Similarly, our quadrature rules for the degree-3 32-node tetrahedron and the degree-4 60-node tetrahedron require 21 and 51 points, respectively, while the quadrature rules currently available in the literature for these elements require 24 and 59 points [24].

4. Error estimates. In this section, we prove that, when conditions C1–C8 are satisfied, the finite element method maintains an optimal order of convergence for a related elliptic problem. Convergence for the wave equation can then be derived in a way analogous to [12, Chapter 4.6].

Throughout this section, we will let p denote the degree of the finite element space, by which we mean the largest degree such that $\tilde{U} \supset \mathcal{P}_p$. We will also let C denote a positive constant that may depend on the domain Ω, the regularity of the mesh, the parameters ρ and c, the reference space $\tilde{U}$, and the reference quadrature rules $(\tilde{\mathcal{Q}}, \{\tilde{\omega}_{\tilde{\mathbf{x}}}\}_{\tilde{\mathbf{x}} \in \tilde{\mathcal{Q}}})$ and $(\tilde{\mathcal{Q}}', \{\tilde{\omega}_{\tilde{\mathbf{x}}}\}_{\tilde{\mathbf{x}} \in \tilde{\mathcal{Q}}'})$ but does not depend on the mesh resolution h and the functions that appear in the inequalities.

4.1. Preliminary results. To obtain error bounds, we first define some norms and function spaces and list a few preliminary results. Let $H^k(\Omega)$ denote the Sobolev space of functions on Ω with order-k square-integrable weak derivatives and equip the space with norm

$$\|u\|_k^2 := \sum_{|\alpha| \le k} \|D^{\boldsymbol{\alpha}} u\|_0^2,$$

where $\| \cdot \|_0$ denotes the L^2-norm, $D^{\boldsymbol{\alpha}} := \partial_1^{\alpha_1} \partial_2^{\alpha_2} \partial_3^{\alpha_3}$ the partial derivative, and $|\boldsymbol{\alpha}| := \alpha_1 + \alpha_2 + \alpha_3$ the order of the derivative. We also define the broken Sobolev spaces

$H^k(\mathcal{T}_h) := \{u \in L^2(\Omega) \mid u|_e \in H^k(e) \text{ for all } e \in \mathcal{T}_h\}$, equipped with norm

$$\|u\|^2_{\mathcal{T}_h,k} := \sum_{e \in \mathcal{T}_h} \|u|_e\|^2_k.$$

Throughout this section, we will use the fact that $H^2(\Omega) \supset \mathcal{C}^0(\overline{\Omega})$ for any 3D domain Ω.

We also define the seminorms $|u|^2_{\mathcal{Q}_h} := (u,u)_{\mathcal{Q}_h}$ and $|\sigma|^2_{\mathcal{Q}'_h} := (\sigma,\sigma)_{\mathcal{Q}'_h}$ for piecewise continuous functions u and σ and define $\Pi_{h,q}$ to be the L^2-projection operators projecting onto the discontinuous piecewise-polynomial spaces $V(\mathcal{T}_h, \mathcal{P}_q) := \{u \in L^2(\Omega) \mid u \circ \phi_e \in \mathcal{P}_q \text{ for all } e \in \mathcal{T}_h\}$. Several useful properties of these spaces and operators are listed below.

LEMMA 4.1. *Let $q \geq 0$. Then*

$$|u_h|_{\mathcal{Q}_h} \leq C\|u_h\|_0 \qquad \text{for all } u_h \in V(\mathcal{T}_h, \mathcal{P}_q),$$
$$|\boldsymbol{\sigma}_h|_{\mathcal{Q}'_h} \leq C\|\boldsymbol{\sigma}_h\|_0 \qquad \text{for all } \boldsymbol{\sigma}_h \in V(\mathcal{T}_h, \mathcal{P}_q)^3.$$

Proof. These results follow immediately from the fact that the elements are affine equivalent with the reference element and that the reference space $\mathcal{P}_q$ is finite dimensional. $\square$

LEMMA 4.2. *If conditions* C1–C4 *are satisfied, then $|\cdot|_{\mathcal{Q}_h}$ becomes a full norm $\|\cdot\|_{\mathcal{Q}_h}$ on $U(\mathcal{T}_h, \tilde{U})$ and*

$$\|u_h\|_{\mathcal{Q}_h} \geq C\|u_h\|_0 \qquad \text{for all } u_h \in U_h.$$

Furthermore, if conditions C1–C3, C6, *and* C7 *are satisfied, then $|\cdot|_{\mathcal{Q}'_h}$ becomes a full norm $\|\cdot\|_{\mathcal{Q}'_h}$ on $V(\mathcal{T}_h, D\tilde{U})$ and*

$$\|\nabla u_h\|_{\mathcal{Q}'_h} \geq C\|\nabla u_h\|_0 \qquad \text{for all } u_h \in U_h.$$

Proof. These inequalities follow immediately from the fact that the elements are affine equivalent with the reference element and that the reference element space $\tilde{U}$ is finite dimensional. $\square$

LEMMA 4.3. *Let $u \in H^k(\mathcal{T}_h)$ and $\boldsymbol{\sigma} \in H^k(\mathcal{T}_h)^3$, with $k \geq 0$, and let $q \geq 0$. Then*

$$\|u - \Pi_{h,q}u\|_{\mathcal{T}_h,m} \leq Ch^{\min(q+1,k)-m}\|u\|_{\mathcal{T}_h,\min(q+1,k)}, \qquad m \leq \min(q+1,k),$$
$$\|\boldsymbol{\sigma} - \Pi_{h,q}\boldsymbol{\sigma}\|_{\mathcal{T}_h,m} \leq Ch^{\min(q+1,k)-m}\|\boldsymbol{\sigma}\|_{\mathcal{T}_h,\min(q+1,k)}, \qquad m \leq \min(q+1,k).$$

Furthermore, if $k \geq 2$, then

$$|u - \Pi_{h,q}u|_{\mathcal{Q}_h} \leq Ch^{\min(q+1,k)}\|u\|_{\mathcal{T}_h,\min(q+1,k)},$$
$$|\boldsymbol{\sigma} - \Pi_{h,q}\boldsymbol{\sigma}|_{\mathcal{Q}'_h} \leq Ch^{\min(q+1,k)}\|\boldsymbol{\sigma}\|_{\mathcal{T}_h,\min(q+1,k)}.$$

Finally, if $u \in H^1(\Omega) \cap H^k(\mathcal{T}_h)$, with $k \geq 2$, then

$$\|u - I_h u\|_{\mathcal{T}_h,m} \leq Ch^{\min(p+1,k)-m}\|u\|_{\mathcal{T}_h,\min(p+1,k)}, \qquad m \leq \min(p+1,k),$$

with $p \geq 2$ the degree of the finite element space.

Proof. The first, second, and last inequalities follow from [4, Chapter 3.1]. For the fourth inequality, let $q^* \geq q$ be a polynomial degree and $\tilde{\mathcal{Q}}^* \supset \tilde{\mathcal{Q}}'$ a set of points such that $\mathcal{P}_{q^*}$ is unisolvent on $\tilde{\mathcal{Q}}^*$. Also, let I_h^* be the interpolation operator that interpolates a function in $H^2(\mathcal{T}_h)$ at the nodes $\mathcal{Q}(\mathcal{T}_h, \tilde{\mathcal{Q}}^*)$ by a function in $V(\mathcal{T}_h, \mathcal{P}_{q^*})$. We can then obtain the fourth inequality as follows:

$$
\begin{aligned}
|\boldsymbol{\sigma} - \Pi_{h,q}\boldsymbol{\sigma}|_{\mathcal{Q}_h'} &= |I_h^*\boldsymbol{\sigma} - \Pi_{h,q}\boldsymbol{\sigma}|_{\mathcal{Q}_h'} \\
&\leq C\|I_h^*\boldsymbol{\sigma} - \Pi_{h,q}\boldsymbol{\sigma}\|_0 \\
&\leq C(\|I_h^*\boldsymbol{\sigma} - \boldsymbol{\sigma}\|_0 + \|\boldsymbol{\sigma} - \Pi_{h,q}\boldsymbol{\sigma}\|_0) \\
&\leq Ch^{\min(q+1,k)}\|\boldsymbol{\sigma}\|_{\mathcal{T}_h, \min(q+1,k)},
\end{aligned}
$$

where we used Lemma 4.1 in the second line and the triangle inequality in the third line. The last line follows from [4, Chapter 3.1].

The third inequality can be derived in a way analogous to the fourth inequality. □

4.2. Estimates on the integration error. Define the two integration errors $r_h(u,w) := (u,w) - (u,w)_{\mathcal{Q}_h}$ and $r_h'(\boldsymbol{\sigma}, \boldsymbol{\tau}) := (\boldsymbol{\sigma}, \boldsymbol{\tau}) - (\boldsymbol{\sigma}, \boldsymbol{\tau})_{\mathcal{Q}_h'}$. In [12] we derived the following bounds on r_h.

LEMMA 4.4. *Let $p \geq 2$ be the degree of the finite element space, $u \in H^k(\Omega)$, with $k \geq 2$, and $w \in U_h$. If conditions C1–C5 are satisfied, then*

$$
\begin{aligned}
|r_h(u,w)| &\leq Ch^{\min(p,k)}\|u\|_{\min(p,k)}\|w\|_1, \\
|r_h(u,w)| &\leq Ch^{\min(p+1,k)}\|u\|_{\min(p+1,k)}\|w\|_{\mathcal{T}_h,2}.
\end{aligned}
$$

We also derive bounds on the integration error for the stiffness matrix.

LEMMA 4.5. *Let $p \geq 2$ be the degree of the finite element space, $\boldsymbol{\sigma} \in H^k(\mathcal{T}_h)^3$ with $k \geq 2$, and $\boldsymbol{\tau} \in V(\mathcal{T}_h, D\tilde{U})^3$. If conditions C1–C3, C6, and C8 are satisfied, then*

$$
(6) \qquad |r_h'(\boldsymbol{\sigma}, \boldsymbol{\tau})| \leq Ch^{\min(p,k)}\|\boldsymbol{\sigma}\|_{\mathcal{T}_h, \min(p,k)}\|\boldsymbol{\tau}\|_0,
$$

$$
(7) \qquad |r_h'(\boldsymbol{\sigma}, \boldsymbol{\tau})| \leq Ch^{\min(p+1,k)}\|\boldsymbol{\sigma}\|_{\mathcal{T}_h, \min(p+1,k)}\|\boldsymbol{\tau}\|_{\mathcal{T}_h,1}.
$$

Proof. Using C8, we can write

$$
r_h'(\boldsymbol{\sigma}, \boldsymbol{\tau}) = r_h'(\boldsymbol{\sigma} - \Pi_{h,p-1}\boldsymbol{\sigma}, \boldsymbol{\tau}).
$$

Inequality (6) then follows from the Cauchy–Schwarz inequality and Lemma 4.3.

Using C8 and the fact that $\mathcal{P}_p \subset \mathcal{P}_{p-1} \otimes D\tilde{U}$ for $p \geq 2$, we can also write

$$
\begin{aligned}
r_h'(\boldsymbol{\sigma}, \boldsymbol{\tau}) &= r_h'\big((\boldsymbol{\sigma} - \Pi_{h,p}\boldsymbol{\sigma}) + (\Pi_{h,p}\boldsymbol{\sigma} - \Pi_{h,p-1}\boldsymbol{\sigma}) + \Pi_{h,p-1}\boldsymbol{\sigma}, (\boldsymbol{\tau} - \Pi_{h,0}\boldsymbol{\tau}) + \Pi_{h,0}\boldsymbol{\tau}\big) \\
&= r_h'(\boldsymbol{\sigma} - \Pi_{h,p}\boldsymbol{\sigma}, \boldsymbol{\tau}) + r_h'(\Pi_{h,p}\boldsymbol{\sigma} - \Pi_{h,p-1}\boldsymbol{\sigma}, \boldsymbol{\tau} - \Pi_{h,0}\boldsymbol{\tau}).
\end{aligned}
$$

Inequality (7) then follows from the Cauchy–Schwarz inequality and Lemma 4.3. □

4.3. Error estimates for a related elliptic problem. Let $v \in \mathcal{C}^0(\overline{\Omega})$. The elliptic problem corresponding to (2) is finding $u \in H_0^1(\Omega)$ such that

$$
(8) \qquad (c\nabla u, \nabla w) = (v, w) \qquad\qquad \text{for all } w \in H_0^1(\Omega).
$$

The corresponding mass-lumped finite element method is finding $u_h \in U_h$ such that

$$
(9) \qquad (c\nabla u_h, \nabla w)_{\mathcal{Q}_h'} = (v, w)_{\mathcal{Q}_h} \qquad\qquad \text{for all } w \in U_h.
$$

In the next two theorems we prove optimal convergence in the H^1-norm and L^2-norm.

THEOREM 4.6 (optimal convergence in the H^1-norm). *Let u be the solution of* (8) *and u_h the solution of* (9). *Assume $c \in \mathcal{C}^p(\overline{\Omega})$, $u \in H^{k_u}(\Omega)$, and $v \in H^{k_v}(\Omega)$, with $k_u, k_v \geq 2$. If conditions* C1–C8 *are satisfied, then*

$$(10) \qquad \|u - u_h\|_1 \leq Ch^{\min(p, k_u - 1, k_v)}(\|u\|_{\min(p+1, k_u)} + \|v\|_{\min(p, k_v)})$$

with $p \geq 2$ the degree of the finite element method.

Proof. Define $e_h := I_h u - u_h$ and $\epsilon_h := u - I_h u$. Using (8), we can write

$$(c\nabla I_h u, \nabla e_h)_{\mathcal{Q}'_h} = -r'_h(c\nabla I_h u, \nabla e_h) - (c\nabla \epsilon_h, \nabla e_h) + (c\nabla u, \nabla e_h)$$
$$= -r'_h(c\nabla I_h u, \nabla e_h) - (c\nabla \epsilon_h, \nabla e_h) + (v, e_h),$$

and using (9), we can obtain

$$(c\nabla u_h, \nabla e_h)_{\mathcal{Q}'_h} = (v, e_h)_{\mathcal{Q}_h}.$$

Subtracting these two equalities gives

$$(11) \qquad (c\nabla e_h, \nabla e_h)_{\mathcal{Q}'_h} = -r'_h(c\nabla I_h u, \nabla e_h) - (c\nabla \epsilon_h, \nabla e_h) + r_h(v, e_h).$$

From Lemma 4.2, the positivity of c, and Poincaré's inequality, it follows that

$$(12) \qquad \|e_h\|_1^2 \leq C(c\nabla e_h, \nabla e_h)_{\mathcal{Q}'_h}.$$

Using Lemma 4.5, Lemma 4.3, and the regularity of c, we can obtain

$$|r'_h(c\nabla I_h u, \nabla e_h)| \leq Ch^{\min(p, k_u - 1)}\|c\nabla I_h u\|_{\mathcal{T}_h, \min(p, k_u - 1)}\|e_h\|_1$$
$$(13) \qquad\qquad\qquad \leq Ch^{\min(p, k_u - 1)}\|u\|_{\min(p+1, k_u)}\|e_h\|_1.$$

Using the Cauchy–Schwarz inequality, the boundedness of c, and Lemma 4.3, we can also obtain

$$(14) \qquad |(c\nabla \epsilon_h, \nabla e_h)| \leq Ch^{\min(p, k_u - 1)}\|u\|_{\min(p+1, k_u)}\|e_h\|_1.$$

Finally, using Lemma 4.4, we can obtain

$$(15) \qquad |r_h(v, e_h)| \leq Ch^{\min(p, k_v)}\|v\|_{\min(p, k_v)}\|e_h\|_1.$$

Combining (11)–(15) gives

$$\|e_h\|_1 \leq Ch^{\min(p, k_u - 1, k_v)}(\|u\|_{\min(p+1, k_u)} + \|v\|_{\min(p, k_v)}).$$

Since $u - u_h = e_h + \epsilon_h$, inequality (10) then follows from the above and Lemma 4.3. □

To prove optimal convergence in the L^2-norm, we make the following regularity assumption: for any $v \in L^2(\Omega)$, the solution of (8) is in $H^2(\Omega)$ and satisfies

$$(16) \qquad \|u\|_2 \leq C\|v\|_0.$$

This is certainly true when $\partial\Omega$ is $\mathcal{C}^2$ and $c \in \mathcal{C}^1(\overline{\Omega})$.

THEOREM 4.7 (optimal convergence in the L^2-norm). *Let u be the solution of* (8) *and u_h the solution of* (9). *Assume $c \in \mathcal{C}^{p+1}(\overline{\Omega})$, $u \in H^{k_u}(\Omega)$, and $v \in H^{k_v}(\Omega)$, with $k_u, k_v \geq 2$, and assume the regularity condition* (16) *holds. If conditions* C1–C8 *are satisfied, then*

$$(17) \qquad \|u - u_h\|_0 \leq Ch^{\min(p+1, k_u, k_v)}(\|u\|_{\min(p+1, k_u)} + \|v\|_{\min(p+1, k_v)})$$

with $p \geq 2$ the degree of the finite element method.

Proof. Define $z_h \in H_0^1(\Omega)$ to be the solution of the elliptic problem

$$(c\nabla z_h, \nabla w) = (u - u_h, w) \qquad \text{for all } w \in H_0^1(\Omega).$$

From the regularity assumption it follows that $z_h \in H^2(\Omega)$ and

$$\|z_h\|_2 \leq C\|u - u_h\|_0.$$

From the definition of z_h, it also follows that

$$
\begin{aligned}
\|u - u_h\|_0^2 &= (c\nabla[u - u_h], \nabla z_h) \\
&= (c\nabla[u - u_h], \nabla[z_h - I_h z_h]) + (c\nabla[u - u_h], \nabla I_h z_h).
\end{aligned}
$$
(18)

We can bound the term $(c\nabla[u - u_h], \nabla[z_h - I_h z_h])$ as follows:

$$
\begin{aligned}
&|(c\nabla[u - u_h], \nabla[z_h - I_h z_h])| \\
&\quad \leq C\|u - u_h\|_1 \|z_h - I_h z_h\|_1 \\
&\quad \leq Ch^{\min(p, k_u-1, k_v)+1}(\|u\|_{\min(p+1,k_u)} + \|v\|_{\min(p,k_v)})\|z_h\|_2 \\
&\quad \leq Ch^{\min(p+1, k_u, k_v+1)}(\|u\|_{\min(p+1,k_u)} + \|v\|_{\min(p,k_v)})\|u - u_h\|_0,
\end{aligned}
$$
(19)

where we used the Cauchy–Schwarz inequality and the boundedness of c in the first line, Theorem 4.6 and Lemma 4.3 in the second line, and the regularity assumption in the last line. It then remains to find a bound for $(c\nabla[u - u_h], \nabla I_h z_h)$.

To do this, use (8) to write

$$(c\nabla u, \nabla I_h z_h) = (v, I_h z_h)$$

and use (9) to write

$$
\begin{aligned}
(c\nabla u_h, \nabla I_h z_h) &= r_h'(c\nabla u_h, \nabla I_h z_h) + (c\nabla u_h, \nabla I_h z_h)_{\mathcal{Q}_h'} \\
&= r_h'(c\nabla u_h, \nabla I_h z_h) + (v, I_h z_h)_{\mathcal{Q}_h}.
\end{aligned}
$$

Subtracting these two equalities gives

$$(c\nabla[u - u_h], \nabla I_h z_h) = -r_h'(c\nabla u_h, \nabla I_h z_h) + r_h(v, I_h z_h).$$
(20)

Now, set $q := \min(p - 1, k_u - 2)$. We can write

$$
\begin{aligned}
r_h'(c\nabla u_h, \nabla I_h z_h) &= r_h'(\nabla u_h, c\nabla I_h z_h - \Pi_{h,0} c\nabla I_h z_h) \\
&= r_h'(\nabla u_h - \Pi_{h,q} \nabla u, c\nabla I_h z_h - \Pi_{h,0} c\nabla I_h z_h) \\
&\quad + r_h'(\Pi_{h,q} \nabla u, c\nabla I_h z_h - \Pi_{h,0} c\nabla I_h z_h) \\
&= r_h'(\nabla u_h - \Pi_{h,q} \nabla u, c\nabla I_h z_h - \Pi_{h,0} c\nabla I_h z_h) + r_h'(c\Pi_{h,q} \nabla u, \nabla I_h z_h) \\
&=: R_1 + R_2,
\end{aligned}
$$

where we used C8 for the first and third equalities. We can bound R_1 as follows:

$$
\begin{aligned}
|R_1| &\leq \|\nabla u_h - \Pi_{h,q} \nabla u\|_0 \|c\nabla I_h z_h - \Pi_{h,0} c\nabla I_h z_h\|_0 \\
&\quad + |\nabla u_h - \Pi_{h,q} \nabla u|_{\mathcal{Q}_h'} |c\nabla I_h z_h - \Pi_{h,0} c\nabla I_h z_h|_{\mathcal{Q}_h'} \\
&\leq Ch\|\nabla u_h - \Pi_{h,q} \nabla u\|_0 \|z_h\|_2 \\
&\leq Ch(\|\nabla u_h - \nabla u\|_0 + \|\nabla u - \Pi_{h,q} \nabla u\|_0)\|u - u_h\|_0 \\
&\leq Ch^{\min(p, k_u-1, k_v)+1}(\|u\|_{\min(p+1,k_u)} + \|v\|_{\min(p,k_v)})\|u - u_h\|_0,
\end{aligned}
$$

where we used the Cauchy–Schwarz inequality for the first inequality, Lemma 4.1, the regularity of c, and Lemma 4.3 for the second inequality, the triangle inequality and the regularity assumption for the third inequality, and Theorem 4.6 and Lemma 4.3 for the last inequality. We can also bound R_2 as follows:

$$
\begin{aligned}
|R_2| &\leq Ch^{p+1}\|c\Pi_{h,q}\nabla u\|_{\mathcal{T}_h,p+1}\|\nabla I_h z_h\|_{\mathcal{T}_h,1} \\
&\leq Ch^{p+1}\|\Pi_{h,q}\nabla u\|_{\mathcal{T}_h,p+1}\|I_h z_h\|_{\mathcal{T}_h,2} \\
&\leq Ch^{p+1}\|\Pi_{h,q}\nabla u\|_{\mathcal{T}_h,p+1}\|z_h\|_2 \\
&= Ch^{p+1}\|\Pi_{h,q}\nabla u\|_{\mathcal{T}_h,q}\|z_h\|_2 \\
&\leq Ch^{p+1}\|\nabla u\|_{\mathcal{T}_h,q}\|z_h\|_2 \\
&\leq Ch^{p+1}\|u\|_{\min(p,k_u-1)}\|u - u_h\|_0.
\end{aligned}
$$

Here, the first line follows from Lemma 4.5 and the fact that $c\Pi_{h,q}\nabla u \in H^{p+1}(\mathcal{T}_h)^3$, the second line follows from the regularity of c, the third line follows from Lemma 4.3, the fifth line follows from Lemma 4.3, and the last line follows from the regularity assumption. The fourth line follows from the fact that $\Pi_{h,q}\nabla u$ is piecewise polynomial of degree q and therefore $\|\Pi_{h,q}\nabla u\|_{\mathcal{T}_h,p+1} = \|\Pi_{h,q}\nabla u\|_{\mathcal{T}_h,q}$. By combining the bounds on R_1 and R_2, we then obtain

$$
\begin{aligned}
|r'_h(c\nabla u_h, \nabla I_h z_h)| = |R_1 + R_2| &\leq |R_1| + |R_2| \\
&\leq Ch^{\min(p+1,k_u,k_v+1)}(\|u\|_{\min(p+1,k_u)} + \|v\|_{\min(p,k_v)})\|u - u_h\|_0.
\end{aligned} \tag{21}
$$

From Lemma 4.4, Lemma 4.3, and the regularity assumption, it also follows that

$$
\begin{aligned}
|r_h(v, I_h z_h)| &\leq Ch^{\min(p+1,k_v)}\|v\|_{\min(p+1,k_v)}\|I_h z\|_{\mathcal{T}_h,2} \\
&\leq Ch^{\min(p+1,k_v)}\|v\|_{\min(p+1,k_v)}\|z_h\|_2 \\
&\leq Ch^{\min(p+1,k_v)}\|v\|_{\min(p+1,k_v)}\|u - u_h\|_0.
\end{aligned} \tag{22}
$$

Combining (20), (21), and (22) gives

$$
|(c\nabla[u - u_h], \nabla I_h z_h)| \leq Ch^{\min(p+1,k_u,k_v)}(\|u\|_{\min(p+1,k_u)} + \|v\|_{\min(p+1,k_v)})\|u - u_h\|_0.
$$

Combining this with (18) and (19) then gives (17).   □

These results can be used to prove optimal convergence for the wave equation in a way analogous to [12, Chapter 4.6] by replacing $a(u, w)$ by $a_h(u, w) := (c\nabla u, \nabla w)_{\mathcal{Q}'_h}$ and by defining the projection operator π_h of [12] such that $a_h(\pi_h u, w) = (\nabla \cdot c\nabla u, w)_{\mathcal{Q}_h}$ for all $w \in U_h$.

4.4. Error estimates for the linear elastic case. So far, we have only analyzed the scalar wave equation, but we can obtain error estimates for the elastic wave equations in an analogous way.

In the linear elastic case, the wave field $\mathbf{u} : \Omega \times (0, T) \to \mathbb{R}^3$ is a vector field and (1a) becomes

$$
\rho\partial_t^2\mathbf{u} = \nabla \cdot C : \nabla\mathbf{u} + \mathbf{f} \qquad\qquad \text{in } \Omega \times (0, T)
$$

with $[\nabla \cdot C : \nabla\mathbf{u}]_i = \sum_{j,k,l=1}^3 \partial_j C_{jilk}\partial_k u_l$, where $C : \Omega \to \mathbb{R}^{3\times3\times3\times3}$ is the elastic tensor field with symmetries $C_{ijkl} = C_{jikl} = C_{ijlk} = C_{klij}$ and bounds

$$
c_0\|\boldsymbol{\sigma} + \boldsymbol{\sigma}^t\| \leq \|C : \boldsymbol{\sigma}\| \leq c_1\|\boldsymbol{\sigma} + \boldsymbol{\sigma}^t\| \qquad\qquad \text{for all } \boldsymbol{\sigma} \in \mathbb{R}^{3\times3}
$$

with c_0, c_1 strictly positive constants and $\|\boldsymbol{\sigma}\|^2 := \sum_{i,j=1}^3 \sigma_{ij}^2$.
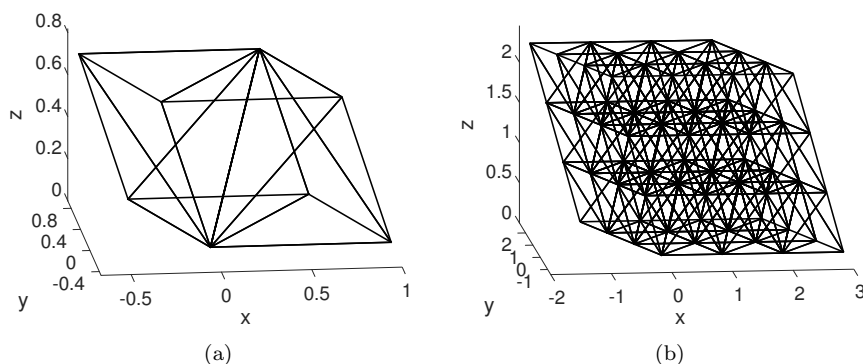
FIG. 1. *Single cell divided into six tetrahedra (left) and repetitions of this cell, resulting in the tetragonal disphenoid honeycomb (right).*

The only part of the error analysis that requires some additional work in this case is the second inequality of Lemma 4.2. Instead of $\|\nabla u\|_{\mathcal{Q}'_h} \geq C\|\nabla u\|_0$, we need to show that if conditions C1–C3, C6, and C7, are satisfied, then

$$(23) \qquad \|\nabla \mathbf{u}_h + \nabla \mathbf{u}_h^t\|_{\mathcal{Q}'_h} \geq C\|\nabla \mathbf{u}_h + \nabla \mathbf{u}_h^t\|_0 \qquad \text{for all } \mathbf{u}_h \in U_h^3.$$

This result follows from the fact that $\tilde{U}$ is finite dimensional and from the relations

$$\|\nabla \mathbf{u}_h + \nabla \mathbf{u}_h^t\|_{\mathcal{Q}'_h}^2 = \sum_{e \in \mathcal{T}_h} \frac{|e|}{|\tilde{e}|} \|\mathbf{J}_e^{-1} \cdot (\tilde{\nabla}\tilde{\mathbf{w}}_e + \tilde{\nabla}\tilde{\mathbf{w}}_e^t) \cdot \mathbf{J}_e^{-t}\|_{\tilde{\mathcal{Q}}'}^2,$$

$$\|\nabla \mathbf{u}_h + \nabla \mathbf{u}_h^t\|_0^2 = \sum_{e \in \mathcal{T}_h} \frac{|e|}{|\tilde{e}|} \|\mathbf{J}_e^{-1} \cdot (\tilde{\nabla}\tilde{\mathbf{w}}_e + \tilde{\nabla}\tilde{\mathbf{w}}_e^t) \cdot \mathbf{J}_e^{-t}\|_{\tilde{e}}^2,$$

where $\mathbf{J}_e := \nabla\phi_e$ is the Jacobian of the element mapping, $\mathbf{J}_e^{-t}$ denotes the transposed of $\mathbf{J}_e^{-1}$, $\tilde{\mathbf{w}}_e := \mathbf{J}_e \cdot (\mathbf{u}_h \circ \phi_e) \in \tilde{U}^3$, and $\|\tilde{\boldsymbol{\sigma}}\|_{\tilde{\mathcal{Q}}'}^2 := \sum_{\tilde{\mathbf{x}} \in \tilde{\mathcal{Q}}'} \omega'_{\tilde{\mathbf{x}}}\|\tilde{\boldsymbol{\sigma}}(\tilde{\mathbf{x}})\|^2.$

Using the boundedness of C, (23), and Korn's inequality, we can show that the bilinear operator for the elastic case $a_h(\mathbf{u}, \mathbf{w}) := (C : \nabla\mathbf{u}, \nabla\mathbf{w})_{\mathcal{Q}'_h}$ is still coercive. The other parts of the error analysis are analogous to the scalar case.

5. Dispersion analysis. To test the effect of the new quadrature rules on the accuracy and time step size, we first analyze the dispersion properties of the resulting mass-lumped finite element method along the lines of [11]. We consider a homogeneous unbounded domain with $\rho = c = 1$ and consider plane waves of the form

$$(24) \qquad\qquad u(\mathbf{x}, t) = e^{\hat{\imath}(\boldsymbol{\kappa} \cdot \mathbf{x} - \omega t)}.$$

Here, $\hat{\imath} := \sqrt{-1}$ denotes the imaginary number, $\boldsymbol{\kappa}$ denotes the wave vector, and ω denotes the angular velocity. We also let $\lambda = 2\pi/|\boldsymbol{\kappa}|$ denote the wavelength and $c_P = \sqrt{c/\rho} = 1$ denote the wave propagation speed. The angular velocity and wave propagation speed satisfy the relation $\omega = |\boldsymbol{\kappa}|c_P$.

To analyze the numerical dispersion, we consider a translation-invariant mesh constructed from a repeated cell pattern as illustrated in Figure 1 and derive the propagation speeds $c_{P,h}$ of the numerical plane waves. The dispersion error e_{disp} is defined as the error in the numerical wave propagation speed: $e_{disp} := |c_P - c_{P,h}|/c_P$.

Since the mesh is constructed from a repeated cell pattern, obtaining the numerical wave propagation speed for a given wave vector requires solving an eigenvalue problem related to only a single cell.

To construct the translation-invariant mesh, we subdivide the unit cell $[0, 1)$ into tetrahedra and repeat this pattern to pack the entire 3D space. We also apply a linear transformation $\mathbf{x} \to \mathbf{T} \cdot \mathbf{x}$, with $\mathbf{T} \in \mathbb{R}^{3 \times 3}$ and $[\mathbf{T} \cdot \mathbf{x}]_i = \sum_{j=1}^{3} T_{ij} x_j$, to the mesh in order to obtain more regular tetrahedra.

Let $\{\mathbf{x}^{(\Omega_0, i)}\}_{i=1}^{n_0}$ denote all the nodes on $\Omega_0 := \mathbf{T} \cdot [0, 1)^3$, let $\{\mathbf{x}^{(\Omega_{\mathbf{k}}, i)}\}_{i=1}^{n_0}$ denote the translated nodes on the translated cell $\Omega_{\mathbf{k}} = \mathbf{T} \cdot \mathbf{k} + \Omega_0$, and let $w^{(\Omega_{\mathbf{k}}, i)}$ denote the corresponding nodal basis functions. We define matrices $M^{(\Omega_0)}, A^{(\Omega_0, \Omega_{\mathbf{k}})} \in \mathbb{R}^{n_0 \times n_0}$ as follows:

$$M_{ij}^{(\Omega_0)} = (\rho w^{(\Omega_0, i)}, w^{(\Omega_0, j)})_{\mathcal{Q}_h}, \qquad i, j = 1, \dots, n_0,$$
$$A_{ij}^{(\Omega_0, \Omega_{\mathbf{k}})} = (c \nabla w^{(\Omega_0, i)}, \nabla w^{(\Omega_0, j)})_{\mathcal{Q}'_h}, \qquad i, j = 1, \dots, n_0, \mathbf{k} \in \{-1, 0, 1\}^3.$$

For any wave vector $\boldsymbol{\kappa} \in \mathbb{R}^3$, we then define the matrix $S^{(\boldsymbol{\kappa})} \in \mathbb{R}^{n_0 \times n_0}$ as follows:

$$S^{(\boldsymbol{\kappa})} = \left(M^{(\Omega_0)} \right)^{-1} \left(\sum_{\mathbf{k} \in \{-1, 0, 1\}^3} e^{\hat{i} (\boldsymbol{\kappa} \cdot \mathbf{T} \cdot \mathbf{k})} A^{(\Omega_0, \Omega_{\mathbf{k}})} \right).$$

When using an order-$2K$ Dablain time integration scheme [10], with time step size Δt, the angular velocities of the numerical plane waves are given by

$$\omega_h^{(\boldsymbol{\kappa}, i)} = \pm \frac{1}{\Delta t} \arccos \left(\sum_{k=0}^{K} \frac{1}{(2k)!} (-\Delta t^2 s_h^{(\boldsymbol{\kappa}, i)})^k \right),$$

where $\{s_h^{(\boldsymbol{\kappa}, i)}\}_{i=1}^{n_0}$ are the eigenvalues of $S^{(\boldsymbol{\kappa})}$ [11]. The numerical wave propagation speeds are given by $c_{P,h}^{(\boldsymbol{\kappa}, i)} = |\omega_h^{(\boldsymbol{\kappa}, i)}| / |\boldsymbol{\kappa}|$. The dispersion error, for a given wavelength λ, is then given by

$$e_{disp}^{(\lambda)} := \sup_{\boldsymbol{\kappa} \in \mathbb{R}^3, |\boldsymbol{\kappa}| = 2\pi/\lambda} \left(\inf_{i=1, \dots, n_0} \frac{|c_P - c_{P,h}^{(\boldsymbol{\kappa}, i)}|}{c_P} \right).$$

For the dispersion analysis, we consider a mesh of congruent nearly regular isofacial tetrahedra, known as the tetragonal disphenoid honeycomb. This mesh is obtained by slicing the unit cube $[0, 1)^3$ into six tetrahedra with the planes $x_1 = x_2$, $x_2 = x_3$, and $x_1 = x_3$, and applying a linear transformation $\mathbf{x} \to \mathbf{T} \cdot \mathbf{x}$, with

$$\mathbf{T} = \begin{bmatrix} 1 & -1/3 & -1/3 \\ 0 & \sqrt{8/9} & -\sqrt{2/9} \\ 0 & 0 & \sqrt{2/3} \end{bmatrix}.$$

An illustration of this mesh is given in Figure 1.

We plot the dispersion error for different elements and quadrature rules against the number of elements per wavelength $N_E := (\lambda^3 / |e|_{av})^{1/3}$, where $|e|_{av} = 2\sqrt{3}/27$ denotes the average element volume. We also compute the largest allowed time step size, given by
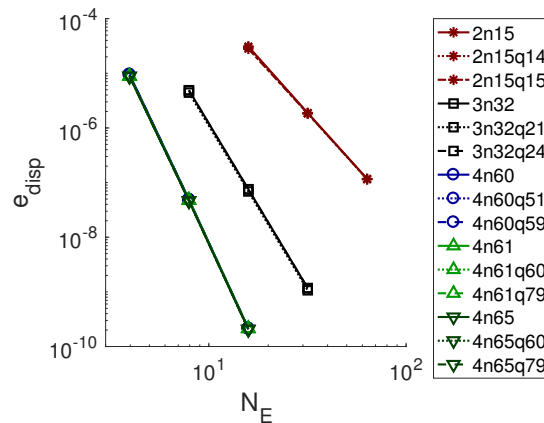
$$\Delta t = \sqrt{c_K / s_{h,\max}},$$

FIG. 2. *Dispersion error e_{disp} versus the number of elements per wave length N_E for different mass-lumped finite element methods. In the legend, $[p]n[n]q[n']$ denotes the degree-p mass-lumped finite element with n nodes and n' quadrature points for evaluating the element stiffness matrix. Solid lines correspond to exact integration, dotted lines correspond to quadrature rules presented in this paper, and dashed lines to quadrature rules taken from [24]. Graph 2n15q15 corresponds to the degree-2 method using the mass matrix quadrature rule as a stiffness matrix quadrature rule. Graphs of methods with the same polynomial degree are almost identical.*

TABLE 7

Dispersion error in terms of the number of elements per wavelength N_E, based on extrapolation of the graphs in Figure 2. The same notation as in the legend of Figure 2 is used. Methods using the new quadrature rules presented in this paper are in bold.

Method	e_{disp}	Method	e_{disp}	Method	e_{disp}
2n15	$1.89(N_E)^{-4}$	4n60	$0.865(N_E)^{-8}$	4n65	$0.825(N_E)^{-8}$
2n15q14	$1.86(N_E)^{-4}$	**4n60q51**	$0.842(N_E)^{-8}$	**4n65q60**	$0.827(N_E)^{-8}$
2n15q15	$1.88(N_E)^{-4}$	4n60q59	$0.861(N_E)^{-8}$	4n65q79	$0.825(N_E)^{-8}$
3n32	$1.20(N_E)^{-6}$	4n61	$0.854(N_E)^{-8}$		
3n32q21	$1.09(N_E)^{-6}$	**4n61q60**	$0.854(N_E)^{-8}$		
3n32q24	$1.19(N_E)^{-6}$	4n61q79	$0.851(N_E)^{-8}$		

with c_K a constant depending on the order of the time integration scheme ($c_K = 4, 12, 7.57, 21.48$ for $K = 1, 2, 3, 4$, respectively) and

$$s_{h,\max} = \sup_{\mathbf{k} \in \mathcal{K}_0} \max_{i=1,\ldots,n_0} s_h^{(\boldsymbol{\kappa},i)}$$

the largest eigenvalue of the discrete spatial operator, with $\mathcal{K}_0 := \mathbf{T}^{-t} \cdot [0, 2\pi)$ the space of distinct wave vectors. Details on the dispersion analysis can be found in [11].

We test the degree-p mass-lumped finite element methods presented in [12] and given in Table 1 using exact stiffness matrix evaluation and using the quadrature rules presented in this paper and quadrature rules that are accurate up to degree $p + p' - 2$ from [24], with p' the highest polynomial degree of the enriched element space. For the degree-2 method, we also test using the quadrature rule of the mass matrix for evaluating the stiffness matrix. We combine each method with an order-$2p$ Dablain time integration scheme.

The dispersion error versus the number of elements per wavelength is shown in Figure 2 and extrapolations of these graphs are given in Table 7. The figure and

TABLE 8
Largest allowed time step size for different mass-lumped finite element methods. The same notation as in the legend of Figure 2 is used. Methods using the new quadrature rules presented in this paper are in bold.

Method	Δt	Method	Δt	Method	Δt
2n15	0.291	4n60	0.0508	4n65	0.0932
2n15q14	0.280	**4n60q51**	0.0580	**4n65q60**	0.0936
2n15q15	0.181	4n60q59	0.0578	4n65q79	0.0935
3n32	0.128	4n61	0.0721		
3n32q21	0.136	**4n61q60**	0.0796		
3n32q24	0.135	4n61q79	0.0792		

table show that the dispersion error of all methods converges with order $2p$, which is typical for eigenvalue and dispersion errors of symmetry-conserving finite element approximations; see, for example, [2] and the references therein. The figure and table also show that there is hardly any difference in accuracy between the methods of the same degree and that the methods using a quadrature rule to evaluate the stiffness matrix have a nearly the same or even slightly smaller dispersion error than the methods using exact integration.

The largest allowed time step size for each method is given in Table 8. The table shows that the quadrature rules for the stiffness matrix tested here hardly affect the largest allowed time step size, except for the degree-2 15-point mass matrix quadrature rule, which reduces the allowed time step size by more than a factor 1.5. For the other quadrature rules, the largest allowed time step size remains nearly the same or becomes even slightly larger.

If the stiffness matrix is evaluated with a quadrature rule and the resulting accuracy and time step size remain nearly the same, then the number of computations to obtain a given accuracy mainly depends on the number of quadrature points. Since the quadrature rules presented in this paper satisfy these properties and require fewer quadrature points than those currently available in the literature, they can result in a reduction of the computational cost proportionate to the reduction in number of quadrature points.

6. Numerical tests.

6.1. Algorithms for computing the element stiffness matrices. Before we present the numerical tests, we first briefly describe the algorithms for computing the element stiffness matrices. In particular, we show how we efficiently compute the element stiffness matrix-vector products on the fly. We do not store the matrices, since this requires storing and fetching significantly more data, and since it was shown in [19] that an on-the-fly approach is more efficient for higher-degree elements.

To describe the algorithms, let $e \in \mathcal{T}_h$ be an arbitrary element. We introduce the following notation:

- $\{\tilde{\mathbf{x}}_i\}_{i=1}^n = \tilde{\mathcal{Q}}$: nodes on reference element $\tilde{e}$. Nodes of the different mass-lumped elements can be found in [12].
- $\tilde{w}_i$: nodal basis function corresponding to $\tilde{\mathbf{x}}_i$.
- $w_i^{(e)} := \tilde{w}_i \circ \phi_e^{-1}$: nodal basis function of the physical element.
- $\{\tilde{\mathbf{x}}_i'\}_{i=1}^{n'} = \tilde{\mathcal{Q}}'$: quadrature points for the stiffness matrix on reference element $\tilde{e}$. Quadrature rules for the different elements are given in section 3.
- $\tilde{\omega}_i'$: quadrature weight corresponding to $\tilde{\mathbf{x}}_i'$.
- $A^{(e)} \in \mathbb{R}^{n \times n}$: the element stiffness matrix.

- $u^{(e)}$: the wave field on e.
- $\underline{u}^{(e)} \in \mathbb{R}^n$: the wave field at the nodes on e.

When using exact integration, the stiffness matrix-vector product $\underline{v}^{(e)} := A^{(e)}\underline{u}^{(e)} \in \mathbb{R}^n$ is given by

$$
(25) \qquad [A^{(e)}\underline{u}^{(e)}]_i = \int_e c\nabla w_i^{(e)} \cdot \nabla u^{(e)} \, dx
$$

for $i = 1, \ldots, n$. After rewriting the integral as an integral over the reference element, this becomes

$$
(26) \qquad [A^{(e)}\underline{u}^{(e)}]_i = \int_{\tilde{e}} \tilde{\nabla}\tilde{w}_i \cdot \tilde{\mathbf{c}}^{(e)} \cdot \tilde{\nabla}\tilde{u}^{(e)} \, d\tilde{x},
$$

where $\tilde{u}^{(e)} := u^{(e)} \circ \phi_e$, $\tilde{\nabla}$ is the gradient operator in reference coordinates, and $\tilde{\mathbf{c}}^{(e)} := (c \circ \phi_e)\frac{|e|}{|\tilde{e}|}\mathbf{J}_e^{-t} \cdot \mathbf{J}_e^{-1}$ is a tensor field, with $\mathbf{J}_e := \nabla\phi_e$ the Jacobian of the element mapping and $\mathbf{J}_e^{-t}$ the transposed of $\mathbf{J}_e^{-1}$. When c is constant within each element, then $\tilde{\mathbf{c}}^{(e)}$ is also constant and we can compute (26) using the algorithm of [19]:

$$
(27) \qquad [A^{(e)}\underline{u}^{(e)}]_i = \sum_{i_D, j_D = 1}^{3} \tilde{c}_{i_D, j_D}^{(e)} \left(\sum_{j=1}^{n} B_{ij}^{(i_D, j_D)} \underline{u}_j^{(e)} \right),
$$

where $B^{(i_D, j_D)} \in \mathbb{R}^{n \times n}$ are precomputed matrices, given by

$$
B_{ij}^{(i_D, j_D)} = \int_{\tilde{e}} (\tilde{\partial}_{i_D}\tilde{w}_i)(\tilde{\partial}_{j_D}\tilde{w}_j) \, d\tilde{x},
$$

for $i_D, j_D = 1, 2, 3$, $i, j = 1, \ldots, n$, with $\tilde{\partial}_{i_D}$ the derivative in reference coordinate i_D. We can reduce the number of computations in (27) using the fact that $\tilde{\mathbf{c}}^{(e)}$ is symmetric:

$$
(28) \qquad [A^{(e)}\underline{u}^{(e)}]_i = \sum_{i_D = 1}^{3} \sum_{j_D = 1}^{i_D} \tilde{c}_{i_D, j_D}^{(e)} \left(\sum_{j=1}^{n} \hat{B}_{ij}^{(i_D, j_D)} \underline{u}_j^{(e)} \right),
$$

where $\hat{B}^{(i_D, j_D)} := B^{(i_D, j_D)} + B^{(j_D, i_D)}$ if $i_D \neq j_D$ and $\hat{B}^{(i_D, j_D)} := B^{(i_D, j_D)}$ when $i_D = j_D$. The complete algorithm can then be described as follows:

A.1. Compute $\epsilon^{(i_D, j_D)} \in \mathbb{R}^n$ for $i_D = 1, 2, 3$, $j_D \leq i_D$:

$$
\epsilon_i^{(i_D, j_D)} = \sum_{j=1}^{n} \hat{B}_{ij}^{(i_D, j_D)} \underline{u}_j^{(e)}.
$$

A.2. Compute $A^{(e)}\underline{u}^{(e)} \in \mathbb{R}^n$:

$$
[A^{(e)}\underline{u}^{(e)}]_i = \sum_{i_D = 1}^{3} \sum_{j_D = 1}^{i_D} \tilde{c}_{i_D, j_D}^{(e)} \epsilon_i^{(i_D, j_D)}.
$$

The computational work is dominated by the first step, where six matrix-vector products with matrices of size $n \times n$ are computed.

Alternatively, we can compute $A^{(e)}\underline{u}^{(e)}$ by evaluating the integrals with a quadrature rule. Equation (26) then becomes

$$(29) \qquad [A^{(e)}\underline{u}^{(e)}]_i = \sum_{k=1}^{n'} \tilde{\nabla}\tilde{w}_i(\tilde{\mathbf{x}}_k') \cdot \tilde{\mathbf{c}}^{(e,k)} \cdot \tilde{\nabla}\tilde{u}^{(e)}(\tilde{\mathbf{x}}_k'),$$

where $\tilde{\mathbf{c}}^{(e,k)} := \tilde{\omega}_k' \tilde{\mathbf{c}}^{(e)}(\tilde{\mathbf{x}}_k') \in \mathbb{R}^{3\times 3}$. We can compute this as follows:

$$(30) \qquad [A^{(e)}\underline{u}^{(e)}]_i = \sum_{k=1}^{n'} \sum_{i_D=1}^{3} D_{ki}^{(i_D)} \left(\sum_{j_D=1}^{3} \tilde{c}_{i_D,j_D}^{(e,k)} \left(\sum_{j=1}^{n} D_{kj}^{(j_D)} \underline{u}_j^{(e)} \right) \right),$$

where $D^{(i_D)} \in \mathbb{R}^{n'\times n}$ are precomputed matrices, given by

$$D_{ki}^{(i_D)} = (\tilde{\partial}_{i_D}\tilde{w}_i)(\tilde{\mathbf{x}}_k').$$

The complete algorithm can be described as follows:

B.1. Compute $\epsilon^{(j_D)} \in \mathbb{R}^{n'}$ for $j_D = 1, 2, 3$:

$$\epsilon_k^{(j_D)} = \sum_{j=1}^{n} D_{kj}^{(j_D)} \underline{u}_j^{(e)}.$$

B.2. Compute $\sigma^{(i_D)} \in \mathbb{R}^{n'}$ for $i_D = 1, 2, 3$:

$$\sigma_k^{(i_D)} = \sum_{j_D=1}^{3} \tilde{c}_{i_D,j_D}^{(e,k)} \epsilon_k^{(j_D)}.$$

B.3. Compute $A^{(e)}\underline{u}^{(e)} \in \mathbb{R}^{n}$:

$$[A^{(e)}\underline{u}^{(e)}]_i = \sum_{i_D=1}^{3} \left(\sum_{k=1}^{n'} D_{ki}^{(i_D)} \sigma_k^{(i_D)} \right).$$

The computational work for this algorithm is dominated by the first and third steps, which both require three matrix-vector products with matrices of size $n' \times n$, so six of these matrix-vector products in total. Since $n' < n$ for all the quadrature rules presented in this paper, this number of computations is smaller than for the previous algorithm, although only slightly. However, as we will show next, this quadrature-based algorithm is significantly more efficient than the exact-integral algorithm in case of linear elasticity. Moreover, this quadrature-based algorithm also works if c varies within the element.

In case of linear elasticity, the wave field $\mathbf{u} : \Omega \times (0, T) \to \mathbb{R}^3$ is a vector field and the term $c\nabla u$ becomes the stress tensor $C : \nabla\mathbf{u}$, with $C \in \mathbb{R}^{3\times 3\times 3\times 3}$ the order-4 elasticity tensor with symmetries $C_{ijkl} = C_{jikl} = C_{ijlk} = C_{klij}$ and $[C : \nabla\mathbf{u}]_{ij} := \sum_{k,l=1}^{3} C_{ijkl}\partial_l u_k$. The vector $\underline{u}^{(e)} \in \mathbb{R}^{3n}$ can in this case be written as a concatenation of three vectors $\underline{u}^{(e,1)}, \underline{u}^{(e,2)}, \underline{u}^{(e,3)} \in \mathbb{R}^n$, where $\underline{u}^{(e,i)}$ is the wave field component u_i at the nodes on e. The parameter $\tilde{\mathbf{c}}^{(e)}$ becomes $\tilde{C}^{(e)} := \frac{|e|}{|\tilde{e}|}\mathbf{J}_e^{-t} \cdot (C \circ \phi_e) \cdot \mathbf{J}_e^{-1}$, where $[\mathbf{J}_e^{-t} \cdot C \cdot \mathbf{J}_e^{-1}]_{ijkl} = \sum_{p,q=1}^{3}[\mathbf{J}_e^{-t}]_{ip}C_{pjkq}[\mathbf{J}_e^{-1}]_{ql}$, and $\tilde{\mathbf{c}}^{(e,k)}$ becomes $\tilde{C}^{(e,k)} := \tilde{\omega}_k'\tilde{C}^{(e)}(\tilde{\mathbf{x}}_k')$. The algorithm for computing the element stiffness matrix-vector product using exact integration then becomes as follows:

A.1. Compute $\epsilon^{(i_D, j_D, j_V)} \in \mathbb{R}^n$ for $i_D, j_D, j_V = 1, 2, 3$:

$$\epsilon_i^{(i_D, j_D, j_V)} = \sum_{j=1}^n B_{ij}^{(i_D, j_D)} \underline{u}_j^{(e, j_V)}.$$

A.2. Define $\underline{v}^{(e)} := A^{(e)} \underline{u}^{(e)}$. Compute $\underline{v}^{(e, i_V)} \in \mathbb{R}^n$ for $i_V = 1, 2, 3$:

$$\underline{v}_i^{(e, i_V)} = \sum_{i_D, j_D, j_V = 1}^3 \tilde{C}_{i_D, i_V, j_V, j_D}^{(e)} \epsilon_i^{(i_D, j_D, j_V)}.$$

The computational work is again dominated by the first step, which now requires 27 matrix-vector products with matrices of size $n \times n$.

When using a quadrature rule, the algorithm becomes as follows:

B.1. Compute $\epsilon^{(j_D, j_V)} \in \mathbb{R}^{n'}$ for $j_D, j_V = 1, 2, 3$:

$$\epsilon_k^{(j_D, j_V)} = \sum_{j=1}^n D_{kj}^{(j_D)} \underline{u}_j^{(e, j_V)}.$$

B.2. Compute $\sigma^{(i_D, i_V)} \in \mathbb{R}^{n'}$ for $i_D, i_V = 1, 2, 3$:

$$\sigma_k^{(i_D, i_V)} = \sum_{j_D, j_V = 1}^3 \tilde{C}_{i_D, i_V, j_V, j_D}^{(e, k)} \epsilon_k^{(j_D, j_V)}.$$

B.3. Define $\underline{v}^{(e)} := A^{(e)} \underline{u}^{(e)}$. Compute $\underline{v}^{(e, i_V)} \in \mathbb{R}^n$ for $i_V = 1, 2, 3$:

$$\underline{v}_i^{(e, i_V)} = \sum_{i_D = 1}^3 \left(\sum_{k=1}^{n'} D_{ki}^{(i_D)} \sigma_k^{(i_D, i_V)} \right).$$

The computational work for this algorithm is dominated again by the first and third steps, which now both require 9 matrix-vector products with matrices of size $n' \times n$, so 18 of these matrix-vector products in total. The number of computations is therefore reduced by more than a factor 1.5 when compared to the algorithm based on exact integration. Furthermore, the quadrature-based algorithm can also handle tensor fields C that vary within the element.

Both algorithms can be slightly improved by exploiting the fact that the rows and columns of the matrices $B^{(i_D, j_D)}$ and the columns of matrices $D^{(i_D)}$ sum to zero. Furthermore, in case of isotropic elasticity, steps A2* and B2* can be computed more efficiently by exploiting the simple structure of the elasticity tensor C.

In the next subsections, we demonstrate the superiority of the quadrature-based algorithm for the case of nonconstant parameters and linear elasticity.

6.2. Acoustic wave on a heterogeneous domain. We first test the methods for an acoustic wave propagation problem with a heterogeneous domain. The acoustic wave equation is given by

$$(31) \qquad \frac{1}{\rho c^2} \partial_t^2 p, = \nabla \cdot \frac{1}{\rho} \nabla p \qquad \qquad \text{in } \Omega \times (0, T),$$

with spatial domain $\Omega \subset \mathbb{R}^3$, time interval $(0, T)$, pressure field $p : \Omega \times (0, T) \to \mathbb{R}$, mass density $\rho : \Omega \to \mathbb{R}$, and acoustic wave speed $c : \Omega \to \mathbb{R}$. We choose $\Omega := (-L_1, L_1) \times (-L_2, L_2) \times (-L_3, L_3)$ and impose zero Neumann boundary conditions.
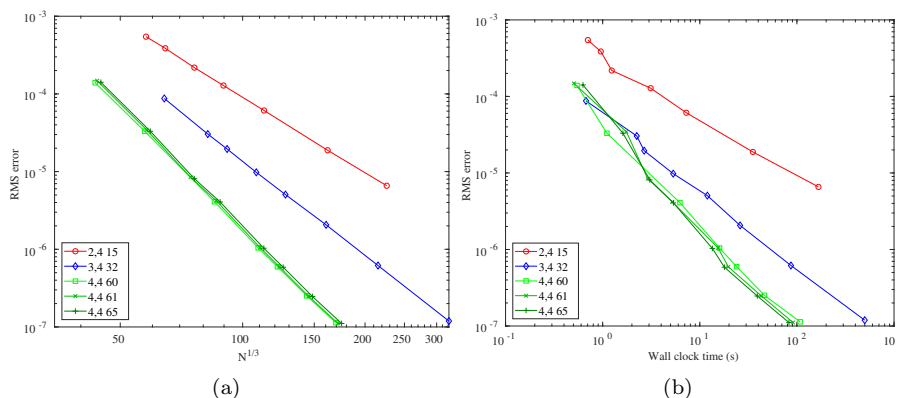
FIG. 3. *RMS errors for the acoustic test case as a function of the cube root of the number of degrees of freedom (left) and as a function of the wall clock time (right). In the legend, the triplet $[p, K\ n]$ refers to the element of degree p with n nodes, combined with an order-K time-stepping scheme. The element stiffness matrices were evaluated using a quadrature rule.*

To construct an analytic solution, let $X_i := x_i + \frac{a_i}{m_i}\cos(m_i x_i)$, for $i = 1, 2, 3$, be distorted coordinates, with $m_i := \frac{1}{2}\pi/L_i$ and $a_i \in [0, 1)$, and define $g_i := \partial_i X_i = 1 - a_i \sin(m_i x_i)$. Also let $\rho_0 \in \mathbb{R}$ be the average mass density, $c_0 \in \mathbb{R}$ the average wave speed, $\mathbf{k} \in \mathbb{R}^3$ the wave vector, and $\omega := c_0|\mathbf{k}|$ the angular velocity, and let parameters ρ and c be given by

$$\rho(\mathbf{x}) := \rho_0 g_1(x_1) g_2(x_2) g_3(x_3), \qquad c(\mathbf{x}) := c_0 \sqrt{\frac{k_1^2 + k_2^2 + k_3^2}{k_1^2 g_1^2(x_1) + k_2^2 g_2^2(x_2) + k_3^2 g_3^2(x_3)}}.$$

Then the standing wave, given by

$$p(\mathbf{x}, t) = \cos(\omega t) \sin(k_1 X_1) \sin(k_2 X_2) \sin(k_3 X_3),$$

is a solution of (31) that satisfies the zero Neumann boundary conditions.

Now, set $L_i = 1$ km, $a_i = 0.2$, $k_i = 3m_i$, for $i = 1, 2, 3$, and $c_0 = 2$ km/s, $\rho_0 = 2$ g/cm^3. To test the numerical methods, we use $p(\mathbf{x}, 0)$ and $\partial_t p(\mathbf{x}, 0)$ as initial conditions. We test on multiple unstructured meshes and simulate in time using a fourth-order time-stepping scheme [10] with time step size $\Delta t = 0.99\Delta t_{\max}$, where $\Delta t_{\max} := \sqrt{12/\sigma_{\max}}$ is the largest allowed time step size [11] and $\sigma_{\max}$ denotes the largest eigenvalue of the spatial operator, which is computed up to four decimals using power iteration. The root mean square (RMS) error is computed after two time oscillations, so at $T = 4\pi/\omega \approx 0.7698$ s.

Figure 3 shows the RMS error plotted against the cube root of the number of degrees of freedom N and the wall clock time for the mass-lumped tetrahedral element methods using the quadrature-based algorithm for the stiffness matrix as discussed in the previous subsection. The simulations shown here were performed with an OpenMP implementation on 24 cores of two Intel Xeon E5-2680 v3 CPUs running at 2.50GHz. Power-law fits of the left graph are also shown in Table 9. This graph shows optimal convergence rates of order $p + 1$ and thereby confirms the error estimates of section 4. In particular, it confirms that optimal convergence rates are maintained, even though the spatial parameters ρ, c vary within the element.

Figure 4 shows the same as Figure 3 for the methods using exact integration to evaluate the stiffness matrix and using a piecewise constant approximation of the

TABLE 9

Power-law fits of the left graphs of Figures 3 *and* 4. *Convergence rates are given in bold.*

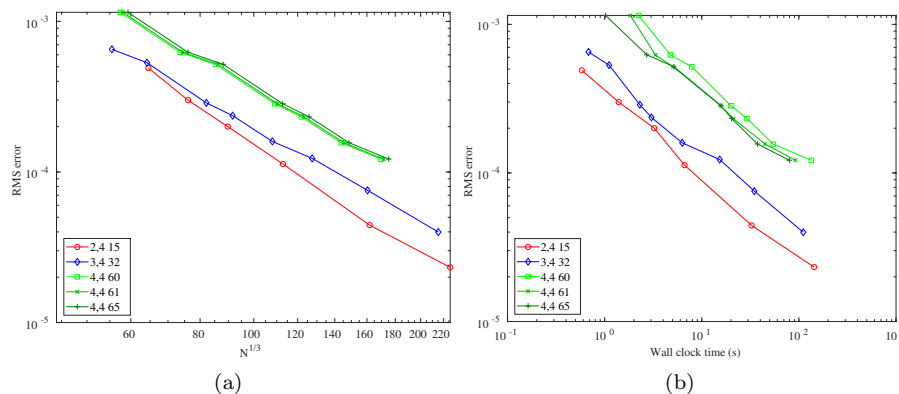|	RMS error	
Method	Figure 3	Figure 4
2,4 15	$(2.9 \times 10^2)N^{(-1/3 \times \mathbf{3.2})}$	$(1.2 \times 10^1)N^{(-1/3 \times \mathbf{2.4})}$
3,4 32	$(1.9 \times 10^3)N^{(-1/3 \times \mathbf{4.1})}$	$(2.8 \times 10^0)N^{(-1/3 \times \mathbf{2.1})}$
4,4 60	$(5.8 \times 10^4)N^{(-1/3 \times \mathbf{5.3})}$	$(5.3 \times 10^0)N^{(-1/3 \times \mathbf{2.1})}$
4,4 61	$(7.9 \times 10^4)N^{(-1/3 \times \mathbf{5.3})}$	$(5.4 \times 10^0)N^{(-1/3 \times \mathbf{2.1})}$
4,4 65	$(7.3 \times 10^4)N^{(-1/3 \times \mathbf{5.3})}$	$(5.7 \times 10^0)N^{(-1/3 \times \mathbf{2.1})}$

FIG. 4. *Same as Figure* 3, *but using exact integration to evaluate the stiffness matrix and using a piecewise constant approximation of the mass density* ρ. *All methods only converge with second order due to the parameter approximation and higher-degree methods only result in more degrees of freedom and computation time.*

mass density ρ. Power-law fits of the left graph are again given in Table 9. The graph shows that, due to the piecewise constant approximation, only second-order convergence rates are obtained. The higher-degree elements now only result in more computations per element, without any significant gain in accuracy. When comparing with Figure 3, it follows that the quadrature-based approach is much more efficient than using exact integration with piecewise constant parameter approximations.

6.3. Elastic wave on a homogeneous domain. We also test the methods for an elastic wave propagation problem on a homogeneous domain. The elastic wave equations are given by

$$\rho \partial_t^2 \mathbf{u} = \nabla \cdot C : \nabla \mathbf{u} + \mathbf{f} \qquad \text{in } \Omega \times (T_0, T_1),$$

with $\mathbf{u} : \Omega \times (T_0, T_1) \to \mathbb{R}^3$ the displacement field, $\mathbf{f} : \Omega \times (T_0, T_1) \to \mathbb{R}^3$ the force field, $\rho : \Omega \to \mathbb{R}$ the mass density, and $C : \Omega \to \mathbb{R}^{3 \times 3 \times 3 \times 3}$ the elasticity tensor. We consider an isotropic elastic medium, so $C : \nabla \mathbf{u} = \lambda(\nabla \cdot \mathbf{u})\mathbf{I} + \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^t)$, with $\mathbf{I} \in \mathbb{R}^{3 \times 3}$ the identity tensor, $\nabla \mathbf{u}^t$ the transposed of $\nabla \mathbf{u}$, and $\lambda, \mu : \Omega \to \mathbb{R}$ the Lamé parameters.

We choose domain $\Omega = [-2, 2] \times [-1, 1] \times [0, 2]$ km^3 with zero Neumann boundary conditions and set the parameters with a constant mass density $\rho = 2$ g/cm^3, primary wave velocity $v_P := \sqrt{(\lambda + 2\mu)/\rho} = 2$ km/s, and secondary/shear wave velocity $w_S := \sqrt{\mu/\rho} = 1.2$ km/s. A unit vertical force source with a 7-Hz Ricker-wavelet is placed at $\mathbf{x}_{src} := (0, 0, 1000)$ m and receivers are placed between $x_r = -1375$ m and
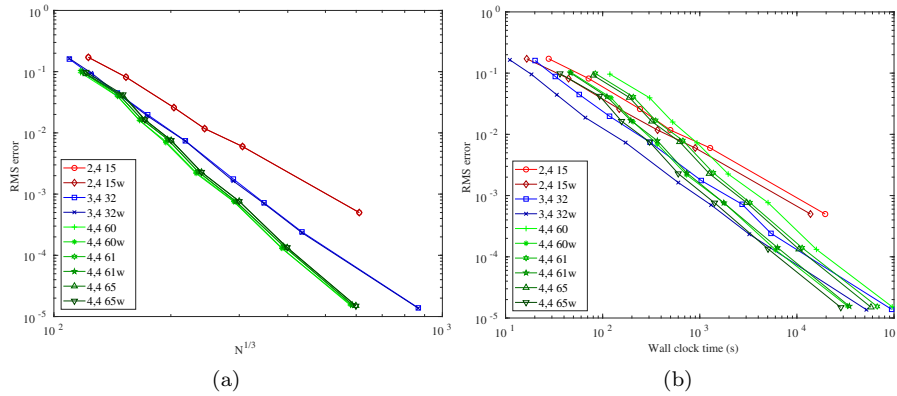
FIG. 5. *RMS errors for the elastic test case as a function of the cube root of the number of scalar degrees of freedom (left) and as a function of the wall clock time (right). In the legend, the triplet [p, K n] refers to the element of degree p with n nodes, combined with an order-K time-stepping scheme. Suffix w denotes elements for which the stiffness matrix was evaluated using the quadrature-based approach described in subsection* 6.1, *while for the other elements we used the exact-integral algorithm.*

TABLE 10

Results for the elastic test case, showing number of scalar degrees of freedom N, number of time steps $N_{\Delta t}$, wall clock time, and RMS error of the degree-p n-node mass-lumped finite element method with stiffness matrix evaluation using exact integration ($n' = -$), a new quadrature rule from this paper (n' bold), or a degree-$(p + p' - 2)$ accurate rule taken from [24], with n' the number of quadrature points.

p	n	n'	N	$N_{\Delta t}$	Time (s)	RMS error
2	15	-	24.6×10^6	366	988	8.24×10^{-3}
		14		371	750	8.24×10^{-3}
3	32	-	10.4×10^6	338	307	7.41×10^{-3}
		21		327	172	7.34×10^{-3}
		24		336	204	7.39×10^{-3}
4	60	-	7.3×10^6	744	725	7.27×10^{-3}
		51		697	366	6.93×10^{-3}
		59		723	407	7.34×10^{-3}
4	61	-	7.6×10^6	631	670	7.89×10^{-3}
		60		614	365	7.87×10^{-3}
		79		614	471	7.88×10^{-3}
4	65	-	8.0×10^6	568	621	7.54×10^{-3}
		60		553	303	7.56×10^{-3}
		79		553	388	7.99×10^{-3}

$x_r = 1375$ m with a 50-m interval at $y_r = 200$ m and $z_r = 800$ m. The exact solution can be found in [1]. The simulation time is chosen such that reflections caused by the boundary conditions do not reach the receivers.

We tested the methods on multiple unstructured meshes and simulated over the time interval $(-0.3, 0.6)$ s, using the time-stepping algorithm as in the previous test case and omitting the initial time steps where the magnitude of the wavelet is smaller than 10^{-16}. Simulations were also carried out in the same environment as in the previous test case. The RMS error is based on the errors at all receivers and for all directional components and is plotted against the cube root of the number of scalar degrees of freedom N and elapsed time in Figure 5. Table 10 also lists the wall

clock time and number of time steps for simulations with an RMS error of around 8×10^{-3} and includes results of simulations where a degree-$(p' + p - 2)$ accurate quadrature rule taken from [24] is used, with p' the highest polynomial degree of the enriched element space. The left graph of Figure 5 confirms that the methods converge with optimal order. It also shows that there is hardly any difference in accuracy between using the quadrature-rule approach or the exact integration algorithm for evaluating the stiffness matrix. Table 10 and the right graph of Figure 5 show that for the degree-3 and degree-4 elements, the quadrature-based algorithm reduces the computational cost by more than a factor 1.5, while for the degree-2 element, this algorithm also results in a moderate speed up. Furthermore, Table 10 also illustrates that the new quadrature rules presented in this paper are more efficient than those currently available in the literature.

7. Conclusion. We presented new and efficient quadrature rules for evaluating the stiffness matrices of mass-lumped tetrahedral elements for wave propagation modeling. These quadrature rules can significantly reduce the number of computations compared to algorithms that evaluate the stiffness matrix using exact integration and can handle spatial parameters that vary within the element without loss of the optimal convergence rate. Obtaining these quadrature rules is not trivial, since degree-p mass-lumped tetrahedral element spaces contain, apart from polynomials up to degree p, numerous additional higher-degree bubble functions when $p \geq 2$. To obtain efficient quadrature rules, we therefore carefully analyzed the stability and accuracy requirements needed to maintain optimal convergence rates. The resulting conditions are presented in this paper, and we prove that if these conditions are met, the resulting method can maintain an optimal order of convergence, even when the spatial parameters vary within the element. We found quadrature rules that satisfy these conditions for recently developed mass-lumped tetrahedral elements of degrees 2 to 4.

For the degree-2 element, the quadrature rule with the smallest number of points we could find was the degree-5 accurate 14-point quadrature rule of [13], but for the degree-3 and degree-4 elements, we found new quadrature rules that require significantly fewer integration points than those currently available. A dispersion analysis shows that by using these quadrature rules, the accuracy and largest allowed time step size remain nearly the same. Several numerical examples also illustrate the accuracy and efficiency of the quadrature-based approach and its superiority to evaluating the integrals for the stiffness matrix exactly. In particular, the quadrature-based approach results in a computational speed-up of around a factor of 1.5 in case of elastic waves. Furthermore, in case of a heterogeneous domain with spatial parameters that vary within the element, the quadrature-based approach results in optimal convergence rates, while exact integration combined with a piecewise constant approximation of the spatial parameters results in a convergence rate of at most order 2.

REFERENCES

[1] K. Aki and P. G. Richards, *Quantitative Seismology, Theory and Methods*, Vol. 1, University Science Books, New York, 1980.

[2] D. Boffi, *Finite element approximation of eigenvalue problems*, Acta Numer., 19 (2010), pp. 1–120.

[3] M. J. S. Chin-Joe-Kong, W. A. Mulder, and M. Van Veldhuizen, *Higher-order triangular and tetrahedral finite elements with mass lumping for solving the wave equation*, J. Engrg. Math., 35 (1999), pp. 405–426.

[4] P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam, 1978.

[5] G. Cohen, P. Joly, and N. Tordjman, *Higher order triangular finite elements with mass lumping for the wave equation*, in Proceedings of the 3rd International Conference on Mathematical and Numerical Aspects of Wave Propagation, SIAM, Philadelphia, 1995, pp. 270–279.

[6] G. C. Cohen, *Higher-Order Numerical Methods for Transient Wave Equations*, Springer, New York, 2002.

[7] G. C. Cohen, P. Joly, J. E. Roberts, and N. Tordjman, *Higher order triangular finite elements with mass lumping for the wave equation*, SIAM J. Numer. Anal., 38 (2001), pp. 2047–2078.

[8] R. Cools, *An encyclopaedia of cubature formulas*, J. Complexity, 19 (2003), pp. 445–453.

[9] T. Cui, W. Leng, D. Lin, S. Ma, and L. Zhang, *High order mass-lumping finite elements on simplexes*, Numer. Math. Theory Methods Appl., 10 (2017), pp. 331–350.

[10] M. Dablain, *The application of high-order differencing to the scalar wave equation*, Geophys., 51 (1986), pp. 54–66.

[11] S. Geevers, W. A. Mulder, and J. J. W. van der Vegt, *Dispersion properties of explicit finite element methods for wave propagation modelling on tetrahedral meshes*, J. Sci. Comput., 77 (2018), pp. 372–396.

[12] S. Geevers, W. A. Mulder, and J. J. W. van der Vegt, *New higher-order mass-lumped tetrahedral elements for wave propagation modelling*, SIAM J. Sci. Comput., 40 (2018), pp. A2801–A2829.

[13] A. Grundmann and H. M. Möller, *Invariant integration formulas for the n-simplex by combinatorial methods*, SIAM J. Numer. Anal., 15 (1978), pp. 282–290.

[14] P. Keast, *Moderate-degree tetrahedral quadrature formulas*, Comput. Methods Appl. Mech. Engrg., 55 (1986), pp. 339–348.

[15] D. Komatitsch and J. P. Vilotte, *The spectral element method: An efficient tool to simulate the seismic response of 2D and 3D geological structures*, Bull. Seismol. Soc. America, 88 (1998), pp. 368–392.

[16] Y. Liu, J. Teng, T. Xu, and J. Badal, *Higher-order triangular spectral element method with optimized cubature points for seismic wavefield modeling*, J. Comput. Phys., 336 (2017), pp. 458–480.

[17] W. A. Mulder, *A comparison between higher-order finite elements and finite differences for solving the wave equation*, in Proceedings of the 2nd ECCOMAS Conference on Numerical Methods in Engineering, John Wiley & Sons, New York, 1996, pp. 344–350.

[18] W. A. Mulder, *New triangular mass-lumped finite elements of degree six for wave propagation*, Progress in Electromagnetics Research, 141 (2013), pp. 671–692.

[19] W. A. Mulder and R. Shamasundar, *Performance of continuous mass-lumped tetrahedral elements for elastic wave propagation with and without global assembly*, Geophys. J. Internat., 207 (2016), pp. 414–421.

[20] A. T. Patera, *A spectral element method for fluid dynamics: Laminar flow in a channel expansion*, J. Comput. Phys., 54 (1984), pp. 468–488.

[21] G. Seriani and E. Priolo, *Spectral element method for acoustic wave simulation in heterogeneous media*, Finite Elem. Anal. Des., 16 (1994), pp. 337–348.

[22] A. H. Stroud, *Approximate Calculation of Multiple Integrals*, Prentice-Hall, Englewood Cliffs, NJ, 1971.

[23] F. D. Witherden and P. E. Vincent, *On the identification of symmetric quadrature rules for finite element methods*, Comput. Math. Appl., 69 (2015), pp. 1232–1241.

[24] L. Zhang, T. Cui, and H. Liu, *A set of symmetric quadrature rules on triangles and tetrahedra*, J. Comput. Math., 27 (2009), pp. 89–96.

[25] E. Zhebel, S. Minisini, A. Kononov, and W. A. Mulder, *A comparison of continuous mass-lumped finite elements with finite differences for 3-D wave propagation*, Geophys. Prospecting, 62 (2014), pp. 1111–1125.