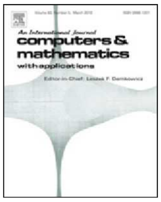
Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa



A locking-free solver for linear elasticity on quadrilateral and hexahedral meshes based on enrichment of Lagrangian elements

Graham Harper [a], Ruishu Wang [b], Jiangguo Liu [a,*], Simon Tavener [a], Ran Zhang [b]

[a] Department of Mathematics, Colorado State University, Fort Collins, CO 80523-1874, USA
[b] School of Mathematical Sciences, Jilin University, Changchun, Jilin 130012, China

ARTICLE INFO

Article history:
Received 15 March 2020
Received in revised form 4 July 2020
Accepted 17 July 2020
Available online xxxx

Keywords:
Enrichment of Lagrangian elements
Hexahedral meshes
Linear elasticity
Locking-free
Quadrilateral meshes
Deal.II implementation

ABSTRACT

This paper presents a new finite element solver for linear elasticity on quadrilateral and hexahedral meshes based on enrichment of the classical bilinear or trilinear Lagrangian elements. It solves the primal variable displacement in the strain–div formulation and can handle both displacement and traction boundary conditions. It is a locking-free solver based on conforming finite elements. The solver has second order accuracy in displacement and first order accuracy in stress and dilation (divergence of displacement), as validated by theoretical analysis and illustrated by numerical experiments on benchmarks. `deal.II` implementation is also discussed.

1. Introduction

This paper concerns finite element methods (FEMs) for linear elasticity prototyped as

$$\begin{cases} -\nabla\cdot\sigma = \mathbf{f}(\mathbf{x}), & \mathbf{x}\in\Omega,\\ \mathbf{u}|_{\Gamma^D} = \mathbf{u}_D, & (\sigma\mathbf{n})|_{\Gamma^N} = \mathbf{t}_N, \end{cases} \tag{1}$$

where Ω is a two- or three-dimensional bounded domain occupied by a homogeneous and isotropic elastic body, $\mathbf{f}$ is a body force, $\mathbf{u}_D$, $\mathbf{t}_N$ are respectively boundary data about displacement and traction posed on the Dirichlet and Neumann boundaries Γ^D and Γ^N, and $\mathbf{n}$ is the outward unit normal vector to the domain boundary $\partial\Omega$. As usual, $\mathbf{u}$ is the solid displacement, $\varepsilon(\mathbf{u}) = \frac{1}{2}\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right)$ is the strain tensor, and $\sigma = 2\mu\,\varepsilon(\mathbf{u}) + \lambda(\nabla\cdot\mathbf{u})\mathbb{I}$ is the Cauchy stress tensor, where $\mathbb{I}$ is the order-two or -three identity matrix. Note that the Lamé constants λ, μ are given by

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \qquad \mu = \frac{E}{2(1+\nu)}, \tag{2}$$

where E is the elasticity modulus and ν is Poisson's ratio. It is known that $\nu \in (0, 0.5)$, so $\lambda \in (0, \infty)$, whereas μ has positive lower and upper bounds.

A challenging issue with development of numerical methods for elasticity is the so-called *Poisson-locking*, which often appears as loss of convergence rates in displacement and/or suspicious behaviors in other quantities such as stress when

[*] Corresponding author.
E-mail addresses: harper@math.colostate.edu (G. Harper), wangrs_math@mail.jlu.edu.cn (R. Wang), liu@math.colostate.edu (J. Liu), tavener@math.colostate.edu (S. Tavener), zhangran@jlu.edu.cn (R. Zhang).

https://doi.org/10.1016/j.camwa.2020.07.014
0898-1221/© 2020 Elsevier Ltd. All rights reserved.

$\lambda \to \infty$ or $\nu \to \frac{1}{2}$, namely, the material becomes nearly incompressible. It is well known that the linear Lagrangian elements Q_1^2 (for rectangles) or P_1^2 (for triangles) suffer Poisson-locking [1,2]. Remedies have been sought [3–6].

There are several approaches for overcoming Poisson-locking.

(i) By design, the mixed finite element methods (MFEMs) based on the Hellinger–Reissner formulation overcome Poisson-locking. These mixed methods consider approximations to stress and displacement simultaneously. The first family of conforming mixed FEMs for triangles was developed in [7]. Analogous finite elements for rectangular and cuboid meshes were developed in [8]. In [9], weakly symmetric conforming mixed FEMs were developed for quadrilaterals. Simple confirming [10] and non-confirming [4] mixed FEMs were developed for rectangles of any dimension. Other related work on MFEMs can be found in [4,11–14]. However, MFEMs involve more unknowns and result in indefinite linear systems that are nontrivial to solve.

(ii) The discontinuous Galerkin (DG) finite element methods provide a different type of choices. The early efforts on interior penalty (IP) DG methods can be found in [15,16]. These IPDG have optimal or suboptimal order convergence in displacement depending on the polynomial types being used. In [17], Nitsche's method was utilized to design IPDG methods for incompressible and nearly incompressible elasticity problems on triangular meshes. These methods are locking-free but introduce too many unknowns. A hybridizable discontinuous Galerkin (HDG) method for linear elasticity was developed in [18]. Approximants are introduced to displacement and stress in element interiors and their traces on the mesh skeleton in the initial formulation of the method. However, after cancellation, "the only globally coupled degrees of freedom are those of an approximation of the displacement defined solely on the faces of the elements".

(iii) The weak Galerkin (WG) methodology offers a new perspective. A family of WG finite element methods have been developed in the primal formulation by using polynomial approximants (degree 1 or higher) on general polygonal or polyhedral meshes [19]. Along this line, [20,21] have developed the lowest order WG methods for linear elasticity. In other words, constant vectors are used separately in element interiors and on interelement boundaries for approximating the displacement.

(iv) Re-using stable Stokes elements for elasticity offers another choice, due to the similarity between Stokes flow and linear elasticity. More details on the MINI elements (on triangles) and the Bernardi–Raugel elements (on triangles and tetrahedra) can be found in [5,22,23].

Finite element approximation on quadrilaterals and hexahedra is important in its own right [24–26]. For practical computation, FEMs on quadrilateral and hexahedral meshes are viable alternatives to FEMs on triangular and tetrahedral meshes, since the former usually involve less degrees of freedom and can still accommodate complicated domain geometry. However, as commented earlier, the classical Lagrangian finite elements are subject to Poisson-locking. Therefore, development of locking-free FEMs for linear elasticity on quadrilateral and hexahedral meshes is an important topic in its own right [6,13,27].

In this paper, we develop a new finite element solver for linear elasticity on quadrilateral and hexahedral meshes based on enrichment of the classical Lagrangian $Q_1^d (d = 2, 3)$ elements. The new method solves the primal variable displacement in the strain–div formulation. The new solver is locking-free, as demonstrated by theoretical analysis and numerical experiments. This new solver for elasticity was motivated by the elasticity discretization on triangles for the Biot's system in [23] and the Bernardi–Raugel element pair on quadrilaterals in [28], but we focus on its computational aspect, especially an accessible `deal.II` implementation [29].

The rest of this paper is organized as follows. In Section 2, we first discuss enrichment of the classical Lagrangian elements and then present our new solver for linear elasticity by utilizing the enriched elements. Section 3 presents error analysis for this new solver and validates its locking-free property. Section 4 discusses implementation of this new solver in `deal.II`. In Section 5, we present numerical results to illustrate its accuracy and locking-free property. Section 6 concludes the paper with comments on other related existing and future work.

2. An elasticity solver based on enriched Lagrangian elements

2.1. Enrichment of Lagrangian elements on quadrilaterals and hexahedra

We first consider convex quadrilaterals in two dimensions. Then we discuss cuboidal hexahedra in three dimensions that have flat faces and hence the outward unit normal vector on each face is a constant vector.

Let E be a convex quadrilateral with vertices $P_i = (x_i, y_i),\ i = 1, 2, 3, 4$. Let e_i be the edge connecting P_{i-1} and $P_i,\ i = 1, 2, 3, 4$. Of course, P_0 is understood as P_4, by the modulo convention. Let $\mathbf{n}_i$ be the outward unit normal vector on edge $e_i,\ i = 1, 2, 3, 4$ (see Fig. 1).

Let $\hat{E} = [0, 1]^2$ (the unit square) be the reference element. Let $(\hat{x}, \hat{y})$ be the reference variables. A bilinear mapping from $\hat{E}$ to E is established as

$$\begin{cases} x = a_1 + a_2\hat{x} + a_3\hat{y} + a_4\hat{x}\hat{y}, \\ y = b_1 + b_2\hat{x} + b_3\hat{y} + b_4\hat{x}\hat{y}, \end{cases} \tag{3}$$

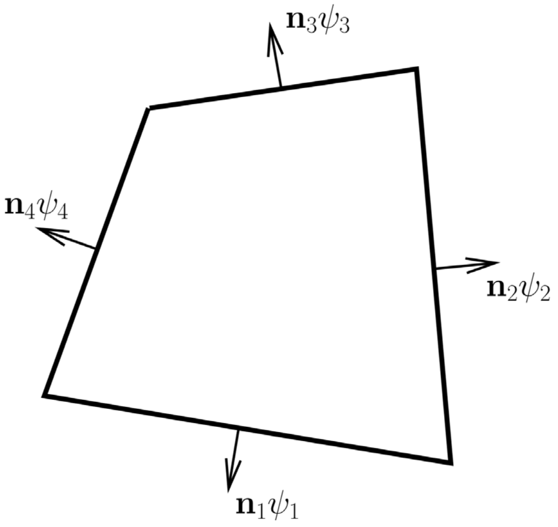


Fig. 1. Four edge-based bubble functions for a quadrilateral.

where the eight coefficients $a_i, b_i (i = 1, 2, 3, 4)$ can be directly calculated using the vertex coordinates:

$$\begin{aligned} &a_1 = x_1, \quad a_2 = x_2 - x_1, \quad a_3 = x_4 - x_1, \quad a_4 = (x_1 + x_3) - (x_2 + x_4), \\ &b_1 = y_1, \quad b_2 = y_2 - y_1, \quad b_3 = y_4 - y_1, \quad b_4 = (y_1 + y_3) - (y_2 + y_4). \end{aligned}$$

We first consider four scalar basis functions defined on $\hat{E}$:

$$\hat{\phi}_1(\hat{x}, \hat{y}) = (1 - \hat{x})(1 - \hat{y}), \qquad \hat{\phi}_2(\hat{x}, \hat{y}) = \hat{x}(1 - \hat{y}), \tag{4}$$

$$\hat{\phi}_3(\hat{x}, \hat{y}) = \hat{x}\hat{y}, \qquad \hat{\phi}_4(\hat{x}, \hat{y}) = (1 - \hat{x})\hat{y}. \tag{5}$$

Let $(x, y) \in E$ be the bilinear mapping image of $(\hat{x}, \hat{y}) \in \hat{E}$. We consider eight vector-valued Lagrangian $Q_1(E)^2$ basis functions on E:

$$\boldsymbol{\phi}_{2i-1}(x, y) = \begin{bmatrix} \hat{\phi}_i(\hat{x}, \hat{y}) \\ 0 \end{bmatrix}, \quad \boldsymbol{\phi}_{2i}(x, y) = \begin{bmatrix} 0 \\ \hat{\phi}_i(\hat{x}, \hat{y}) \end{bmatrix}, \quad i = 1, 2, 3, 4. \tag{6}$$

Now we consider four more scalar basis functions defined on $\hat{E}$:

$$\hat{\psi}_1(\hat{x}, \hat{y}) = \hat{x}(1 - \hat{x})(1 - \hat{y}), \tag{7}$$

$$\hat{\psi}_2(\hat{x}, \hat{y}) = \hat{x}(1 - \hat{y})\hat{y}, \tag{8}$$

$$\hat{\psi}_3(\hat{x}, \hat{y}) = \hat{x}(1 - \hat{x})\hat{y}, \tag{9}$$

$$\hat{\psi}_4(\hat{x}, \hat{y}) = (1 - \hat{x})(1 - \hat{y})\hat{y}, \tag{10}$$

and accordingly four vector-valued edge-based bubble functions on E:

$$\boldsymbol{\psi}_i(x, y) = \mathbf{n}_i\, \psi_i(x, y) = \mathbf{n}_i\, \hat{\psi}_i(\hat{x}, \hat{y}), \quad i = 1, 2, 3, 4. \tag{11}$$

The classical Lagrangian space $Q_1(E)^2$ is now enriched as

$$\mathrm{EQ}_1(E) = \mathrm{Q}_1(E)^2 + \mathrm{Span}(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \boldsymbol{\psi}_3, \boldsymbol{\psi}_4). \tag{12}$$

Next we consider a cuboidal hexahedron E with six flat faces that is treated as a trilinear mapping image of the reference unit cube $\hat{E} = [0, 1]^3$, whose six faces are identified as follows.

#1	#2	#3	#4	#5	#6
$\hat{x} = 0$ Left	$\hat{x} = 1$ Right	$\hat{y} = 0$ Back	$\hat{y} = 1$ Front	$\hat{z} = 0$ Bottom	$\hat{z} = 1$ Top

The trilinear mapping is expressed as

$$\mathbf{p} = (x, y, z) = \mathbf{p}_{000} + \mathbf{p}_a\hat{x} + \mathbf{v}_b\hat{y} + \mathbf{v}_c\hat{z} + \mathbf{p}_d\hat{y}\hat{z} + \mathbf{p}_e\hat{z}\hat{x} + \mathbf{p}_f\hat{x}\hat{y} + \mathbf{p}_g\hat{x}\hat{y}\hat{z}, \tag{13}$$

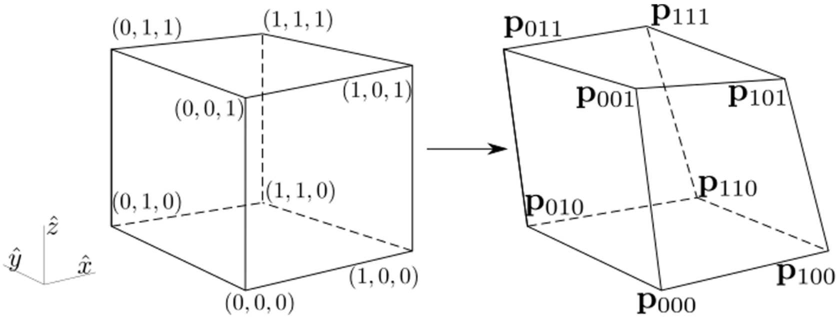


Fig. 2. An illustration for the trilinear mapping for a general hexahedron.

where the points and vectors are defined as

$$\begin{aligned}
&\mathbf{p}_a = \mathbf{p}_{100} - \mathbf{p}_{000}, && \mathbf{p}_d = (\mathbf{p}_{011} - \mathbf{p}_{000}) - (\mathbf{p}_b + \mathbf{p}_c),\\
&\mathbf{p}_b = \mathbf{p}_{010} - \mathbf{p}_{000}, && \mathbf{p}_e = (\mathbf{p}_{101} - \mathbf{p}_{000}) - (\mathbf{p}_c + \mathbf{p}_a),\\
&\mathbf{p}_c = \mathbf{p}_{001} - \mathbf{p}_{000}, && \mathbf{p}_f = (\mathbf{p}_{110} - \mathbf{p}_{000}) - (\mathbf{p}_a + \mathbf{p}_b),\\
&\mathbf{p}_g = (\mathbf{p}_{111} - \mathbf{p}_{000}) - ((\mathbf{p}_a + \mathbf{p}_b + \mathbf{p}_c) + (\mathbf{p}_d + \mathbf{p}_e + \mathbf{p}_f))
\end{aligned}$$

and further explained in Fig. 2. It is interesting to note that the hexahedron E is a parallelepiped if and only if the vector $\mathbf{p}_g$ vanishes.

For the six faces of $\hat{E} = [0, 1]^3$, we define six scalar-valued bubble functions:

$$\begin{cases}
\hat{\psi}_1 = \hat{y}(1-\hat{y})\hat{z}(1-\hat{z})(1-\hat{x}), & \hat{\psi}_2 = \hat{y}(1-\hat{y})\hat{z}(1-\hat{z})\hat{x},\\
\hat{\psi}_3 = \hat{x}(1-\hat{x})\hat{z}(1-\hat{z})(1-\hat{y}), & \hat{\psi}_4 = \hat{x}(1-\hat{x})\hat{z}(1-\hat{z})\hat{y},\\
\hat{\psi}_5 = \hat{x}(1-\hat{x})\hat{y}(1-\hat{y})(1-\hat{z}), & \hat{\psi}_6 = \hat{x}(1-\hat{x})\hat{y}(1-\hat{y})\hat{z}.
\end{cases} \tag{14}$$

Suppose E is a cuboidal hexahedron with six flat faces and $\mathbf{n}_i (1 \le i \le 6)$ are the constant outward unit normal vectors on these faces. Similarly, we define six face-based vector-valued bubble functions as follows

$$\boldsymbol{\psi}_i(x, y, z) = \mathbf{n}_i\, \psi_i(x, y, z) = \mathbf{n}_i\, \hat{\psi}_i(\hat{x}, \hat{y}, \hat{z}), \quad 1 \le i \le 6. \tag{15}$$

Then the classical Lagrangian space $Q_1(E)^3$ for a cuboidal hexahedron E is enriched as

$$\mathrm{EQ}_1(E) = Q_1(E)^3 + \mathrm{Span}(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \boldsymbol{\psi}_3, \boldsymbol{\psi}_4, \boldsymbol{\psi}_5, \boldsymbol{\psi}_6). \tag{16}$$

Remark. For a general hexahedron, some faces may not be flat. If the distortion is not too significant, then we can use the average of the outward normal vectors or that at the face center for a non-flat face in Eq. (15).

2.2. A new finite element scheme for elasticity based on enrichment of Lagrangian elements

In this subsection, we develop a finite element scheme for solving linear elasticity on quadrilateral and hexahedral meshes based on enrichment of the bilinear or trilinear Lagrangian elements.

Let Ω be a polygonal or polyhedral domain equipped with a shape-regular convex quadrilateral/hexahedral mesh $\mathcal{E}_h$. Let $\mathbf{V}_h$ be the space spanned by the basis functions discussed in the previous subsection. Let $\mathbf{V}_h^0$ be the subspace of $\mathbf{V}_h$ consisting of those shape functions that vanish on Γ^D. For $\mathbf{v} \in \mathbf{V}_h$, it is known that $\nabla \cdot \mathbf{v}$ is not a constant on each element E. We shall consider its elementwise average $\overline{\nabla \cdot \mathbf{v}}$.

For the elasticity boundary value problem (1), we now establish a finite element scheme in the strain–div formulation as follows. Find $\mathbf{u}_h \in \mathbf{V}_h$ such that $\mathbf{u}_h|_{\Gamma_h^D} = \Pi_D \mathbf{u}_D$ and

$$\mathcal{A}_h(\mathbf{u}_h, \mathbf{v}) = \mathcal{F}_h(\mathbf{v}), \qquad \forall \mathbf{v} \in \mathbf{V}_h^0, \tag{17}$$

where

$$\mathcal{A}_h(\mathbf{u}_h, \mathbf{v}) = 2\mu \sum_{E \in \mathcal{E}_h} (\varepsilon(\mathbf{u}_h), \varepsilon(\mathbf{v}))_E + \lambda \sum_{E \in \mathcal{E}_h} (\overline{\nabla \cdot \mathbf{u}_h}, \overline{\nabla \cdot \mathbf{v}})_E, \tag{18}$$

$$\mathcal{F}_h(\mathbf{v}) = \sum_{E \in \mathcal{E}_h} (\mathbf{f}, \mathbf{v})_E + \sum_{e \in \Gamma_h^N} \langle \mathbf{t}_N, \mathbf{v} \rangle_e, \tag{19}$$

and

$$(\Pi_D \mathbf{u}_D)|_e = \tilde{\Pi}_D \mathbf{u}_D + \left(\int_e (\mathbf{u}_D - \tilde{\Pi}_D \mathbf{u}_D) \cdot \mathbf{n} \Big/ \int_e \boldsymbol{\psi}_e \cdot \mathbf{n} \right) \boldsymbol{\psi}_e, \quad e \in \Gamma_h^D, \tag{20}$$

with $\tilde{\Pi}_D$ being the interpolation operator into $Q_1(e)^d$ and $\boldsymbol{\psi}_e \in \{\boldsymbol{\psi}_i\}_{i=1}^{2d}$ being the basis functions that satisfy $(\boldsymbol{\psi}_i)|_e \neq \mathbf{0}$.

Remarks.

(i) Compared to the usual numerical scheme using the classical Lagrangian Q_1^d elements, now the finite element space $\mathbf{V}_h$ used in Scheme (17)–(19) is larger.

(ii) In (18), the elementwise average of the dilation is used. In other words, the dilation is approximated by piecewise constants. Note that the enlarged finite element space is the same as the BR_1 space used in [28] for approximating the velocity in Stokes problems, which corresponds to the piecewise constant space for approximating the pressure.

Enforcing boundary conditions. Clearly, there are two sets of basis functions: node-based and edge/face-based. Compatibility among these two types of functions needs to be maintained in enforcement or incorporation of boundary conditions. Here we discuss treatments for three-dimensional problems only, since two-dimensional treatment is similar.

(i) For a Dirichlet face, one can directly enforce the displacement boundary condition at its four vertices. Then the difference between the original Dirichlet data and the interpolant is utilized to calculate the coefficient for the face bubble function.

(ii) For a Neumann face, integrals of the traction data against the thirteen basis functions (twelve for the four vertices of the face, one for the face itself) are computed directly and used independently to determine the coefficients for the respective basis functions.

Computation of numerical stress σ_h. This will be done elementwise. Again the elementwise average of dilation is used, as shown below

$$\sigma_h = 2\mu\, \varepsilon(\mathbf{u}_h) + \lambda\, \overline{\nabla \cdot \mathbf{u}_h}\, \mathbb{I}. \tag{21}$$

3. Analysis: Accuracy and locking-free property

This section validates the *locking-free* property of the proposed finite element scheme while providing error analysis for the scheme. For ease of presentation, we consider problem (1) with a homogeneous Dirichlet boundary condition on the whole boundary.

Definition 1 (*Local and Global Projection Operators*)**.** Let E be a quadrilateral or cuboidal hexahedron with vertices $P_i, i = 1, 2, \dots, 2^d$ ($d = 2$ or 3). We define a projection operator $\Pi_E : H^1(E)^d \to \mathrm{EQ}_1(E)$ as

$$\Pi_E \mathbf{v} = \tilde{\Pi}_E \mathbf{v} + \sum_{i=1}^{2d} \alpha_i \boldsymbol{\psi}_i, \tag{22}$$

with

$$\tilde{\Pi}_E \mathbf{v} = \sum_{i=1}^{2^d} \mathbf{v}(P_i)\, \phi_i \tag{23}$$

and

$$\alpha_i = \left(\int_{e_i} (\mathbf{v} - \tilde{\Pi}_E \mathbf{v}) \cdot \mathbf{n}_i \right) \Big/ \int_{e_i} \boldsymbol{\psi}_i \cdot \mathbf{n}_i, \qquad 1 \le i \le 2d. \tag{24}$$

Note that here $\phi_i, 1 \le i \le 2^d$ are the classical scalar-valued Lagrangian basis functions in $Q_1(E)$ for a quadrilateral or hexahedron. Then we define the global projection operator $\Pi_h : H^1(\Omega)^d \to \mathbf{V}_h$ by

$$(\Pi_h \mathbf{v})|_E = \Pi_E(\mathbf{v}|_E). \tag{25}$$

Accordingly, $\Pi_h \mathbf{v} \in \mathbf{V}_h^0$ if $\mathbf{v} \in H_0^1(\Omega)^d$.

Lemma 1 (*Properties of the Global Projection Operator*)**.** *For any $E \in \mathcal{E}_h$, the following properties hold true*

$$(\Pi_h \mathbf{v})(P_i) = \mathbf{v}(P_i), \quad 1 \le i \le 2^d, \tag{26}$$

$$\overline{\nabla \cdot (\mathbf{v} - \Pi_h \mathbf{v})}|_E = 0. \tag{27}$$

Proof. The scalar-valued Lagrangian basis functions $\phi_i, i = 1, \dots, 2^d$ and vector-valued basis functions $\boldsymbol{\psi}_i, i = 1, \dots, 2d$ defined in Section 2 satisfy

$$\begin{aligned} \phi_i(P_j) &= \delta_{i,j}, \quad i, j = 1, \dots, 2^d, \\ \boldsymbol{\psi}_i(P_j) &= \mathbf{0}, \quad i = 1, \dots, 2d, \quad j = 1, \dots, 2^d, \\ \boldsymbol{\psi}_i|_{e_j} &= \mathbf{0}, \quad i, j = 1, \dots, 2d, \quad i \neq j. \end{aligned}$$

Since $\boldsymbol{\psi}_i(P_j) = \mathbf{0}$ and $(\tilde{\Pi}_E\mathbf{v})(P_i) = \mathbf{v}(P_i)$, we have $(\Pi_E\mathbf{v})(P_i) = \mathbf{v}(P_i)$. It follows from the definition of Π_E that

$$\int_{e_i} (\mathbf{v} - \Pi_E\mathbf{v}) \cdot \mathbf{n}_i = \int_{e_i} (\mathbf{v} - \tilde{\Pi}_E\mathbf{v}) \cdot \mathbf{n}_i - \int_{e_i} \sum_{j=1}^{2d} \alpha_j \boldsymbol{\psi}_j \cdot \mathbf{n}_i.$$

Combined with the fact that $\boldsymbol{\psi}_j|_{e_i} = \mathbf{0}$ $(i \neq j)$, this implies

$$\begin{aligned} &\int_{e_i} (\mathbf{v} - \Pi_E\mathbf{v}) \cdot \mathbf{n}_i \\ &= \int_{e_i} (\mathbf{v} - \tilde{\Pi}_E\mathbf{v}) \cdot \mathbf{n}_i - \int_{e_i} \alpha_i \boldsymbol{\psi}_i \cdot \mathbf{n}_i \\ &= \int_{e_i} (\mathbf{v} - \tilde{\Pi}_E\mathbf{v}) \cdot \mathbf{n}_i - \alpha_i \int_{e_i} \boldsymbol{\psi}_i \cdot \mathbf{n}_i \\ &= \int_{e_i} (\mathbf{v} - \tilde{\Pi}_E\mathbf{v}) \cdot \mathbf{n}_i - \left(\int_{e_i} (\mathbf{v} - \tilde{\Pi}_E\mathbf{v}) \cdot \mathbf{n}_i \Big/ \int_{e_i} \boldsymbol{\psi}_i \cdot \mathbf{n}_i \right) \int_{e_i} \boldsymbol{\psi}_i \cdot \mathbf{n}_i \\ &= 0. \end{aligned}$$

Therefore,

$$\begin{aligned} \|\overline{\nabla \cdot (\mathbf{v} - \Pi_E\mathbf{v})}\|^2 &= \sum_{E \in \mathcal{E}_h} \overline{\nabla \cdot (\mathbf{v} - \Pi_E\mathbf{v})} \int_E \nabla \cdot (\mathbf{v} - \Pi_E\mathbf{v}) \\ &= \sum_{E \in \mathcal{E}_h} \overline{\nabla \cdot (\mathbf{v} - \Pi_E\mathbf{v})} \int_{\partial E} (\mathbf{v} - \Pi_E\mathbf{v}) \cdot \mathbf{n} \\ &= 0, \end{aligned}$$

which completes the proof. □

Definition 2 (*Energy Norm or h-norm*)**.** Since $\mathbf{V}_h^0$ is a subspace of $H_0^1(\Omega)^d$, the Korn's inequality holds [1,30] with a constant $C > 0$ that is independent of λ and h:

$$\|\mathbf{v}_h\|_1 \leq C\|\varepsilon(\mathbf{v}_h)\|, \qquad \forall \mathbf{v}_h \in \mathbf{V}_h^0.$$

And hence we can define a norm on the finite element space $\mathbf{V}_h^0$ by

$$\| \cdot \|_h^2 = \mathcal{A}_h(\cdot, \cdot). \tag{28}$$

Lemma 2 (*1st Order Convergence in Energy Norm*)**.** *Let $\mathbf{u} \in H^2(\Omega)^d$ be the exact solution of (1) and $\mathbf{u}_h \in \mathbf{V}_h^0$ be the numerical solution obtained from the finite element scheme (17) with a homogeneous Dirichlet boundary condition. Then*

$$\|\mathbf{u} - \mathbf{u}_h\|_h \leq Ch\|\mathbf{f}\|, \tag{29}$$

where $C > 0$ is a constant that is independent of λ, h.

Proof. Note that the regularity estimate established in [3] for the elasticity boundary value problem (1) with a homogeneous Dirichlet boundary condition implies that

$$\|\mathbf{u}\|_2 + \lambda\|\nabla \cdot \mathbf{u}\|_1 \leq C\|\mathbf{f}\|. \tag{30}$$

Note that there exists one constant $C > 0$ such that for $\mathbf{v} \in H^1(\Omega)^d$,

$$\|\nabla \cdot \mathbf{v}\| \leq C\|\varepsilon(\mathbf{v})\| \leq C\|\nabla\mathbf{v}\| \leq C\|\mathbf{v}\|_1. \tag{31}$$

It follows from the definition of Π_h that for any constant q on $E \in \mathcal{E}_h$,

$$(\nabla \cdot \mathbf{v} - \nabla \cdot \Pi_h\mathbf{v},\ q)_E = 0. \tag{32}$$

It is known [31] that the error between $\mathbf{u}$ and $\mathbf{u}_h$ could be estimated as

$$\|\mathbf{u}-\mathbf{u}_h\|_h \le 2 \inf_{\mathbf{w}_h\in\mathbf{V}_h} \|\mathbf{u}-\mathbf{w}_h\|_h + \sup_{\mathbf{0}\neq\mathbf{w}_h\in\mathbf{V}_h^0} \frac{|\mathcal{A}_h(\mathbf{u}-\mathbf{u}_h,\mathbf{w}_h)|}{\|\mathbf{w}_h\|_h}. \tag{33}$$

It follows from (31), (32), and the projection properties that

$$\begin{aligned}
&2\mu(\varepsilon(\mathbf{u}-\Pi_h\mathbf{u}),\varepsilon(\mathbf{u}-\Pi_h\mathbf{u}))_E + \lambda(\nabla\cdot(\mathbf{u}-\Pi_h\mathbf{u}),\overline{\nabla\cdot(\mathbf{u}-\Pi_h\mathbf{u})})_E\\
\le\ & 2\mu\|\varepsilon(\mathbf{u}-\Pi_h\mathbf{u})\|_E\|\varepsilon(\mathbf{u}-\Pi_h\mathbf{u})\|_E + 0\\
\le\ & C\|\mathbf{u}-\Pi_h\mathbf{u}\|_1\|\varepsilon(\mathbf{u}-\Pi_h\mathbf{u})\|_E\\
\le\ & Ch\|\mathbf{u}\|_{H^2(E)}\|\varepsilon(\mathbf{u}-\Pi_h\mathbf{u})\|_E.
\end{aligned}$$

We sum the above estimate over the mesh and apply cancellation to obtain

$$\|\mathbf{u}-\Pi_h\mathbf{u}\|_h \le Ch\|\mathbf{u}\|_2.$$

Applying the regularity estimate (30), we obtain

$$\|\mathbf{u}-\Pi_h\mathbf{u}\|_h \le Ch\|\mathbf{f}\|. \tag{34}$$

For the first term on the right side of (33), we use $\Pi_h\mathbf{u}\in\mathbf{V}_h$ to obtain

$$\inf_{\mathbf{w}_h\in\mathbf{V}_h}\|\mathbf{u}-\mathbf{w}_h\|_h \le \|\mathbf{u}-\Pi_h\mathbf{u}\|_h \le Ch\|\mathbf{f}\|.$$

Next, we take a detour to handle $\mathbf{w}_h \in \mathbf{V}_h$. We use (31), the definition of the energy norm, and the fact that μ has a positive lower bound to obtain

$$\|\nabla\cdot\mathbf{w}_h\| \le C\|\varepsilon(\mathbf{w}_h)\| \le C\|\mathbf{w}_h\|_h.$$

Now we return to the 2nd term on the right side of (33). Note $\overline{\nabla\cdot\mathbf{u}}$ denotes the elementwise average. We use (17), the projection properties, the Cauchy–Schwarz inequality, the above estimate, the approximation capacity, and the regularity estimate (30) to obtain

$$\begin{aligned}
&\sup_{\mathbf{0}\neq\mathbf{w}_h\in\mathbf{V}_h^0}\frac{|\mathcal{A}_h(\mathbf{u}-\mathbf{u}_h,\mathbf{w}_h)|}{\|\mathbf{w}_h\|_h} = \sup_{\mathbf{0}\neq\mathbf{w}_h\in\mathbf{V}_h^0}\frac{|\mathcal{A}_h(\mathbf{u},\mathbf{w}_h)-\mathcal{F}_h(\mathbf{w}_h)|}{\|\mathbf{w}_h\|_h}\\
=\ &\sup_{\mathbf{0}\neq\mathbf{w}_h\in\mathbf{V}_h^0}\frac{\left|\lambda(\overline{\nabla\cdot\mathbf{u}}-\nabla\cdot\mathbf{u},\nabla\cdot\mathbf{w}_h)\right|}{\|\mathbf{w}_h\|_h} \le Ch\lambda\|\nabla\cdot\mathbf{u}\|_1 \le Ch\|\mathbf{f}\|,
\end{aligned}$$

which yields (29). □

Theorem 1 (*1st Order Convergence in Stress and Dilation*). *Let $\mathbf{u}\in H^2(\Omega)^d$ be the exact solution of (1) and $\mathbf{u}_h\in\mathbf{V}_h^0$ be the numerical solution obtained from the finite element scheme (17) with a homogeneous Dirichlet boundary condition. Let σ_h be the numerical stress recovered from (21). Then*

$$\lambda\|\nabla\cdot\mathbf{u}-\overline{\nabla\cdot\mathbf{u}_h}\| \le Ch\|\mathbf{f}\|, \tag{35}$$

$$\|\sigma-\sigma_h\| \le Ch\|\mathbf{f}\|, \tag{36}$$

where $C>0$ is a constant that is independent of λ, h.

Proof. The variational formulation for the elasticity problem (1) implies

$$2\mu\sum_{E\in\mathcal{E}_h}(\varepsilon(\mathbf{u}),\varepsilon(\mathbf{v}))_E + \lambda\sum_{E\in\mathcal{E}_h}(\nabla\cdot\mathbf{u},\nabla\cdot\mathbf{v})_E = \mathcal{F}_h(\mathbf{v}),\quad \forall\mathbf{v}\in\mathbf{V}_h^0. \tag{37}$$

For the numerical solution $\mathbf{u}_h$, the FE scheme (17) can be rewritten as

$$2\mu\sum_{E\in\mathcal{E}_h}(\varepsilon(\mathbf{u}_h),\varepsilon(\mathbf{v}))_E + \lambda\sum_{E\in\mathcal{E}_h}(\overline{\nabla\cdot\mathbf{u}_h},\overline{\nabla\cdot\mathbf{v}})_E = \mathcal{F}_h(\mathbf{v}),\quad \forall\mathbf{v}\in\mathbf{V}_h^0. \tag{38}$$

Subtracting (38) from (37) gives

$$\lambda\sum_{E\in\mathcal{E}_h}(\nabla\cdot\mathbf{u}-\overline{\nabla\cdot\mathbf{u}_h},\nabla\cdot\mathbf{v})_E = -2\mu\sum_{E\in\mathcal{E}_h}(\varepsilon(\mathbf{u}-\mathbf{u}_h),\varepsilon(\mathbf{v}))_E. \tag{39}$$

Recall $\overline{\nabla\cdot\mathbf{u}}$ is an elementwise average (see Lemma 2 proof). From the fact that $\mathbf{u}|_{\partial\Omega}=\mathbf{0}$, we obtain

$$\int_\Omega \overline{\nabla\cdot\mathbf{u}} = \sum_{E\in\mathcal{E}_h}\int_E\overline{\nabla\cdot\mathbf{u}} = \sum_{E\in\mathcal{E}_h}\int_E\nabla\cdot\mathbf{u} = \int_\Omega\nabla\cdot\mathbf{u} = \int_{\partial\Omega}\mathbf{u}\cdot\mathbf{n} = 0,$$

which implies $\overline{\nabla\cdot\mathbf{u}} \in L_0^2(\Omega)$. Similarly, $\mathbf{u}_h|_{\partial\Omega} = \mathbf{0}$ implies $\overline{\nabla\cdot\mathbf{u}_h} \in L_0^2(\Omega)$. Thus, $\overline{\nabla\cdot\mathbf{u}} - \overline{\nabla\cdot\mathbf{u}_h} \in L_0^2(\Omega)$. By [1] Lemma 11.2.3, there exists $\mathbf{w} \in H_0^1(\Omega)^d$ such that

$$\nabla\cdot\mathbf{w} = \overline{\nabla\cdot\mathbf{u}} - \overline{\nabla\cdot\mathbf{u}_h}, \tag{40}$$

$$\|\mathbf{w}\|_1 \le \|\overline{\nabla\cdot\mathbf{u}} - \overline{\nabla\cdot\mathbf{u}_h}\|. \tag{41}$$

Now we use (39) and (32) with $\mathbf{v} = \Pi_h\mathbf{w} \in \mathbf{V}_h^0$ to obtain

$$\begin{aligned}
&\lambda\|\overline{\nabla\cdot\mathbf{u}} - \overline{\nabla\cdot\mathbf{u}_h}\|_E^2\\
&= \lambda(\overline{\nabla\cdot\mathbf{u}} - \overline{\nabla\cdot\mathbf{u}_h}, \overline{\nabla\cdot\mathbf{u}} - \overline{\nabla\cdot\mathbf{u}_h})_E\\
&= \lambda(\overline{\nabla\cdot\mathbf{u}} - \overline{\nabla\cdot\mathbf{u}_h}, \nabla\cdot\mathbf{w})_E\\
&= \lambda(\overline{\nabla\cdot\mathbf{u}} - \overline{\nabla\cdot\mathbf{u}_h}, \nabla\cdot\Pi_h\mathbf{w})_E\\
&= \lambda(\nabla\cdot\mathbf{u} - \overline{\nabla\cdot\mathbf{u}_h}, \nabla\cdot\Pi_h\mathbf{w})_E - \lambda(\nabla\cdot\mathbf{u} - \overline{\nabla\cdot\mathbf{u}}, \nabla\cdot\Pi_h\mathbf{w})_E\\
&= 2\mu(\varepsilon(\mathbf{u}_h - \mathbf{u}), \varepsilon(\Pi_h\mathbf{w}))_E - \lambda(\nabla\cdot\mathbf{u} - \overline{\nabla\cdot\mathbf{u}}, \nabla\cdot\Pi_h\mathbf{w})_E.
\end{aligned} \tag{42}$$

Applying the Cauchy–Schwarz inequality, (31), and the projection properties, we obtain

$$\begin{aligned}
&2\mu(\varepsilon(\mathbf{u}_h - \mathbf{u}), \varepsilon(\Pi_h\mathbf{w}))_E\\
&\le C\|\varepsilon(\mathbf{u}_h - \mathbf{u})\|_E\|\Pi_h\mathbf{w}\|_{H^1(E)}\\
&\le C\|\varepsilon(\mathbf{u}_h - \mathbf{u})\|_E(\|\Pi_h\mathbf{w} - \mathbf{w}\|_{H^1(E)} + \|\mathbf{w}\|_{H^1(E)})\\
&\le C\|\varepsilon(\mathbf{u}_h - \mathbf{u})\|_E\|\mathbf{w}\|_{H^1(E)}.
\end{aligned} \tag{43}$$

Similarly, we have

$$\lambda(\nabla\cdot\mathbf{u} - \overline{\nabla\cdot\mathbf{u}}, \nabla\cdot\Pi_h\mathbf{w})_E \le C\lambda h\|\nabla\cdot\mathbf{u}\|_{H^1(E)}\|\mathbf{w}\|_{H^1(E)}. \tag{44}$$

Applying (42), (43), and (44) on the whole mesh, we obtain

$$\begin{aligned}
&\lambda\|\overline{\nabla\cdot\mathbf{u}} - \overline{\nabla\cdot\mathbf{u}_h}\|^2 = \sum_{E\in\mathcal{E}_h}\lambda\|\overline{\nabla\cdot\mathbf{u}} - \overline{\nabla\cdot\mathbf{u}_h}\|_E^2\\
&\le \sum_{E\in\mathcal{E}_h}2\mu(\varepsilon(\mathbf{u}_h - \mathbf{u}), \varepsilon(\Pi_h\mathbf{w}))_E - \sum_{E\in\mathcal{E}_h}\lambda(\nabla\cdot\mathbf{u} - \overline{\nabla\cdot\mathbf{u}}, \nabla\cdot\Pi_h\mathbf{w})_E\\
&\le \sum_{E\in\mathcal{E}_h}C\|\varepsilon(\mathbf{u}_h - \mathbf{u})\|_E\|\mathbf{w}\|_{H^1(E)} + \sum_{E\in\mathcal{E}_h}C\lambda h\|\nabla\cdot\mathbf{u}\|_{H^1(E)}\|\mathbf{w}\|_{H^1(E)}\\
&\le C(\|\mathbf{u}_h - \mathbf{u}\|_h + h\lambda\|\nabla\cdot\mathbf{u}\|_1)\|\mathbf{w}\|_1\\
&\le C(\|\mathbf{u}_h - \mathbf{u}\|_h + h\lambda\|\nabla\cdot\mathbf{u}\|_1)\|\overline{\nabla\cdot\mathbf{u}} - \overline{\nabla\cdot\mathbf{u}_h}\|,
\end{aligned}$$

which gives, by cancellation,

$$\lambda\|\overline{\nabla\cdot\mathbf{u}} - \overline{\nabla\cdot\mathbf{u}_h}\| \le C(\|\mathbf{u}_h - \mathbf{u}\|_h + h\lambda\|\nabla\cdot\mathbf{u}\|_1).$$

By a triangle inequality and approximation capacity, we have

$$\begin{aligned}
\lambda\|\nabla\cdot\mathbf{u} - \overline{\nabla\cdot\mathbf{u}_h}\| &\le \lambda\|\nabla\cdot\mathbf{u} - \overline{\nabla\cdot\mathbf{u}}\| + \lambda\|\overline{\nabla\cdot\mathbf{u}} - \overline{\nabla\cdot\mathbf{u}_h}\|\\
&\le C(h\lambda\|\nabla\cdot\mathbf{u}\|_1 + \|\mathbf{u}_h - \mathbf{u}\|_h) \le Ch\|\mathbf{f}\|.
\end{aligned}$$

For the last step, the regularity of the exact solution has also been used.

Finally, by (21), (31), (35), (30), and the projection properties, we have

$$\begin{aligned}
\|\sigma - \sigma_h\| &= \|2\mu\,\varepsilon(\mathbf{u}) + \lambda(\nabla\cdot\mathbf{u})\mathbb{I} - 2\mu\,\varepsilon(\mathbf{u}_h) - \lambda\overline{\nabla\cdot\mathbf{u}_h}\mathbb{I}\|\\
&\le 2\mu\|\varepsilon(\mathbf{u} - \mathbf{u}_h)\| + d\lambda\|\nabla\cdot\mathbf{u} - \overline{\nabla\cdot\mathbf{u}_h}\|\\
&\le C\|\mathbf{u} - \mathbf{u}_h\|_1 + Ch\|\mathbf{f}\|\\
&\le Ch\|\mathbf{u}\|_2 + Ch\|\mathbf{f}\|\\
&\le Ch\|\mathbf{f}\|,
\end{aligned}$$

which proves (36). □

Theorem 2 (*2nd Order Convergence in Displacement*)**.** *Let $\mathbf{u} \in H^2(\Omega)^d$ be the exact solution of (1) and $\mathbf{u}_h \in \mathbf{V}_h^0$ be the numerical solution obtained from the finite element scheme (17) with a homogeneous Dirichlet boundary condition. Then*

$$\|\mathbf{u} - \mathbf{u}_h\| \le Ch^2\|\mathbf{f}\|, \tag{45}$$

where $C > 0$ is a constant that is independent of λ, h.

Proof. We apply a duality argument. Assume $\boldsymbol{\zeta} \in H^2(\Omega)^d$ is the solution to the following dual problem

$$\begin{aligned} -\nabla \cdot (2\mu\varepsilon(\boldsymbol{\zeta}) + \lambda(\nabla \cdot \boldsymbol{\zeta})\mathbf{I}) &= \mathbf{u} - \mathbf{u}_h, \quad \text{in } \Omega, \\ \boldsymbol{\zeta} &= \mathbf{0}, \quad \text{on } \partial\Omega. \end{aligned}$$

It follows from [3] that

$$\|\boldsymbol{\zeta}\|_2 + \lambda\|\nabla \cdot \boldsymbol{\zeta}\|_1 \leq C\|\mathbf{u} - \mathbf{u}_h\|. \tag{46}$$

Using the fact that $(\mathbf{u} - \mathbf{u}_h)|_{\partial\Omega} = \mathbf{0}$, we obtain

$$\begin{aligned} \|\mathbf{u} - \mathbf{u}_h\|^2 &= (-\nabla \cdot (2\mu\varepsilon(\boldsymbol{\zeta}) + \lambda(\nabla \cdot \boldsymbol{\zeta})\mathbf{I}), \mathbf{u} - \mathbf{u}_h) \\ &= \mathcal{A}(\boldsymbol{\zeta}, \mathbf{u} - \mathbf{u}_h), \end{aligned}$$

where for convenience we have defined

$$\mathcal{A}(\boldsymbol{\zeta}, \mathbf{u} - \mathbf{u}_h) = 2\mu(\varepsilon(\boldsymbol{\zeta}), \varepsilon(\mathbf{u} - \mathbf{u}_h)) + \lambda(\nabla \cdot \boldsymbol{\zeta}, \nabla \cdot (\mathbf{u} - \mathbf{u}_h)).$$

Next, we apply the usual technique to split $\|\mathbf{u} - \mathbf{u}_h\|^2$ as three terms:

$$\begin{aligned} \|\mathbf{u} - \mathbf{u}_h\|^2 &= \big(\mathcal{A}(\boldsymbol{\zeta}, \mathbf{u} - \mathbf{u}_h) - \mathcal{A}_h(\boldsymbol{\zeta}, \mathbf{u} - \mathbf{u}_h)\big) \\ &\quad + \mathcal{A}_h(\boldsymbol{\zeta} - \Pi_h\boldsymbol{\zeta}, \mathbf{u} - \mathbf{u}_h) + \mathcal{A}_h(\Pi_h\boldsymbol{\zeta}, \mathbf{u} - \mathbf{u}_h) \\ &=: T_1 + T_2 + T_3 \end{aligned} \tag{47}$$

and conduct estimation for each term.

For term T_1, we apply (29), (46), and the projection properties to obtain

$$\begin{aligned} &\mathcal{A}(\boldsymbol{\zeta}, \mathbf{u} - \mathbf{u}_h) - \mathcal{A}_h(\boldsymbol{\zeta}, \mathbf{u} - \mathbf{u}_h) \\ &= 2\mu\,(\varepsilon(\boldsymbol{\zeta}), \varepsilon(\mathbf{u} - \mathbf{u}_h)) + \lambda\,(\nabla \cdot \boldsymbol{\zeta}, \nabla \cdot (\mathbf{u} - \mathbf{u}_h)) \\ &\quad -2\mu\,(\varepsilon(\boldsymbol{\zeta}), \varepsilon(\mathbf{u} - \mathbf{u}_h)) - \lambda\,(\overline{\nabla \cdot \boldsymbol{\zeta}}, \overline{\nabla \cdot (\mathbf{u} - \mathbf{u}_h)}) \\ &= (\lambda(\nabla \cdot \boldsymbol{\zeta} - \overline{\nabla \cdot \boldsymbol{\zeta}}), \nabla \cdot (\mathbf{u} - \mathbf{u}_h)) \\ &\leq Ch\lambda\|\nabla \cdot \boldsymbol{\zeta}\|_1\|\varepsilon(\mathbf{u} - \mathbf{u}_h)\| \\ &\leq Ch\lambda\|\nabla \cdot \boldsymbol{\zeta}\|_1\|\mathbf{u} - \mathbf{u}_h\|_h \\ &\leq Ch^2\|\mathbf{f}\|\,\|\mathbf{u} - \mathbf{u}_h\|. \end{aligned} \tag{48}$$

For term T_2, we apply (29), (46), and the projection properties again to obtain

$$\begin{aligned} &\mathcal{A}_h(\boldsymbol{\zeta} - \Pi_h\boldsymbol{\zeta}, \mathbf{u} - \mathbf{u}_h) \\ &= 2\mu(\varepsilon(\boldsymbol{\zeta} - \Pi_h\boldsymbol{\zeta}), \varepsilon(\mathbf{u} - \mathbf{u}_h)) + \lambda(\overline{\nabla \cdot (\boldsymbol{\zeta} - \Pi_h\boldsymbol{\zeta})}, \overline{\nabla \cdot (\mathbf{u} - \mathbf{u}_h)}) \\ &\leq Ch\|\boldsymbol{\zeta}\|_2\|\mathbf{u} - \mathbf{u}_h\|_h \\ &\leq Ch^2\|\mathbf{f}\|\,\|\mathbf{u} - \mathbf{u}_h\|. \end{aligned} \tag{49}$$

To estimate T_3, we test (1) with $\Pi_h\boldsymbol{\zeta}$ and use $(\Pi_h\boldsymbol{\zeta})|_{\partial\Omega} = \mathbf{0}$ to derive

$$\mathcal{F}_h(\Pi_h\boldsymbol{\zeta}) = 2\mu(\varepsilon(\mathbf{u}), \varepsilon(\Pi_h\boldsymbol{\zeta})) + \lambda(\nabla \cdot \mathbf{u}, \nabla \cdot (\Pi_h\boldsymbol{\zeta})).$$

Then we combine (30), (46), and the projection properties to obtain

$$\begin{aligned} &\mathcal{A}_h(\Pi_h\boldsymbol{\zeta}, \mathbf{u} - \mathbf{u}_h) \\ &= \mathcal{A}_h(\Pi_h\boldsymbol{\zeta}, \mathbf{u}) - \mathcal{F}_h(\Pi_h\boldsymbol{\zeta}) \\ &= \lambda(\overline{\nabla \cdot \mathbf{u}}, \overline{\nabla \cdot (\Pi_h\boldsymbol{\zeta})}) - \lambda(\nabla \cdot \mathbf{u}, \nabla \cdot (\Pi_h\boldsymbol{\zeta})) \\ &= \lambda((\overline{\nabla \cdot \mathbf{u}} - \nabla \cdot \mathbf{u}), \nabla \cdot (\Pi_h\boldsymbol{\zeta}) - \overline{\nabla \cdot (\Pi_h\boldsymbol{\zeta})}) \\ &\leq Ch^2\lambda\|\nabla \cdot \mathbf{u}\|_1\|\boldsymbol{\zeta}\|_2 \\ &\leq Ch^2\|\mathbf{f}\|\,\|\mathbf{u} - \mathbf{u}_h\|. \end{aligned} \tag{50}$$

Finally a combination of (47)–(50) leads to the desired result (45). □

Remarks. For Theorems 1 and 2, if we consider general boundary conditions, then the right-hand sides of these error estimates should be revised to

$$\|\mathbf{f}\| + \|\mathbf{u}_D\|_{\mathbf{H}^{\frac{3}{2}}(\Gamma^D)} + \|\mathbf{t}_N\|_{\mathbf{H}^{\frac{1}{2}}(\Gamma^N)}.$$

For a particular problem, each of these three terms may or may not depend on λ. These norms may go large or stay bounded as λ goes large.

4. Implementation in deal.II

The new solver developed in this paper applies to linear elasticity problems in both two and three dimensions. Interestingly, it can be implemented in a dimension-independent way on the deal.II platform. In addition, we have also Matlab implementation for its two-dimensional version and the code modules are included in our own package DarcyLite. This provides additional mutual verification for our code implementation.

deal.II is a popular C++ finite element package for solving partial differential equations [29]. It uses quadrilateral and hexahedral meshes but does not support simplicial (triangular or tetrahedral) meshes. The former involve less degrees of freedom in general. If needed, we can refine a triangular or tetrahedral mesh into a quadrilateral or hexahedral mesh, respectively [32].

deal.II supports a dimension-independent programming paradigm through use of template functions, so implementation of the enriched Lagrangian Q_1 elements (also named as BR_1 elements) was accomplished through writing a new shape function class and finite element class

```
PolynomialsBernardiRaugel<dim>
FE_BernardiRaugel<dim>
```

using template functions with dimension as a template parameter. The finite element class is a derived class of

```
FE_PolyTensor<PolynomialType, dim, spacedim>
```

We will refer to these classes without their template parameters for convenience. This finite element was released as a part of deal.II Version 9.1.

On the reference square, each shape function in (6) is constructed as a tensor product of univariate Lagrangian shape functions in each component. This allows us to take advantage of the class Polynomials::LagrangeEquidistant, from which we define two polynomials $p_0(t) = 1 - t$ and $p_1(t) = t$ so that

$$\begin{aligned} \hat{\phi}_1(\hat{x},\hat{y}) &= p_0(\hat{x})p_0(\hat{y}), \qquad \hat{\phi}_2(\hat{x},\hat{y}) = p_1(\hat{x})p_0(\hat{y}), \\ \hat{\phi}_3(\hat{x},\hat{y}) &= p_1(\hat{x})p_1(\hat{y}), \qquad \hat{\phi}_4(\hat{x},\hat{y}) = p_0(\hat{x})p_1(\hat{y}). \end{aligned}$$

Here we refer to the shape functions in the same clockwise order for consistency with (6), but deal.II orders vertices and faces lexicographically. Each of the polynomials ϕ_i are multiplied by $\mathbb{R}^2$ standard basis vectors $[1,0]^T$ or $[0,1]^T$ to yield the $\boldsymbol{\phi}_i$ shape functions for Q_1^2. The construction of Q_1^3 basis functions is similar.

The bubble functions are constructed via the

```
Polynomials::LagrangeEquidistant
```

class as well, but with the addition of a third polynomial $p_2(t) = 4t(1-t)$. The bubble functions in (11) are constructed without the factor 4 so that

$$\begin{aligned} \hat{\psi}_1(\hat{x},\hat{y}) &= p_2(\hat{x})p_0(\hat{y}), \qquad \hat{\psi}_2(\hat{x},\hat{y}) = p_1(\hat{x})p_2(\hat{y}), \\ \hat{\psi}_3(\hat{x},\hat{y}) &= p_2(\hat{x})p_1(\hat{y}), \qquad \hat{\psi}_4(\hat{x},\hat{y}) = p_0(\hat{x})p_2(\hat{y}). \end{aligned}$$

In deal.II, these are arranged in the lexicographical order of faces and normalized to a maximum value of 1. Each shape function is then multiplied by the corresponding normal vector. This construction has been extended to the unit cube also.

The unit square and cube are mapped to quadrilaterals and hexahedra via bilinear and trilinear mappings, see, e.g., Fig. 2 and Eq. (13). The shape functions discussed in Section 2.1 are evaluated using the mappings from the unit square and cube to quadrilaterals and hexahedra accordingly. The vector fields for bubble functions on the distorted mesh in Example 4 are mapped to the physical cell via the Piola transformation; however, we plan to update deal.II to override the fill_fe_values function. This would involve calling the base implementation in the FE_PolyTensor class by the derived FE_BernardiRaugel class and then adjusting the bubble function vector field to use the normal vector of the physical cell so that

$$\psi_i(\mathbf{x}) = \vec{\mathbf{n}}_i \hat{\psi}_i(\hat{x}).$$

This further improvement is planned to appear in deal.II Version 9.3.

Computation of finite element quantities, e.g., local stiffness matrices, involves the Jacobian matrix and determinant, especially the transpose of the inverse of the Jacobian matrix. The Jacobian determinant and quadrature weights are combined by deal.II into a single function call for simplicity, denoted as JxW, since it depends on the quadrature points only. This reduces the work required by the user when computing the discrete bilinear form. The following code computes elementwise averages of divergence of displacement.

```
for (unsigned int i = 0; i < dofs_per_cell; ++i)
 for (unsigned int q_i = 0; q_i < n_q_points; ++q_i)
  {
   avg_div[i] += fe_values[displacement].divergence(i,q_i)
          *fe_values.JxW(q_i) / cell->measure();
  }
```

The following code computes the bilinear form for the elasticity problem.

```
for (unsigned int q_i = 0; q_i < n_q_points; ++q_i)
 for (unsigned int i = 0; i < dofs_per_cell; ++i)
  for (unsigned int j = 0; j < dofs_per_cell; ++j)
   {
    const Tensor<2, dim> grad_phi_i =
     fe_values[displacement].symmetric_gradient(i,q_i);
    const Tensor<2, dim> grad_phi_j =
     fe_values[displacement].symmetric_gradient(j,q_i);
    cell_matrix(i,j) +=
     (2. *Ex.mu *scalar_product(grad_phi_i,grad_phi_j)
      + Ex.lambda *avg_div[i] *avg_div[j])
     *fe_values.JxW(q_i);
   }
```

The bilinear form for our elasticity solver based on the enriched Lagrangian elements is symmetric positive-definite (SPD). The global discrete system is therefore solved by a conjugate-gradient type linear solver in `deal.II` after the constrained degrees of freedom are condensed out of the sparse system. Multigrid preconditioning methods have potential to reduce the linear solver iterations, but this will be a topic of future investigation.

Data visualization is accomplished by postprocessing the linear system coefficients and then projecting them into the $Q_0(E)^d$ space. The postprocessed data is then saved by `deal.II` to a `.vtk` file that can be visualized with `ParaView` or `VisIt`. For this paper, we used `VisIt`.

5. Numerical experiments

This section presents numerical experiments to demonstrate accuracy and robustness of our new solver for linear elasticity.

Example 1 (*Locking-Free*)**.** This example is directly taken from [19] (Section 9.3 therein). It has also been tested in [20,21]. Specifically, the domain is $\Omega = (0,1)^2$, a Neumann boundary condition is posed on the right boundary of the domain, whereas the other three sides have Dirichlet boundary conditions derived from a known analytical solution for displacement

$$\mathbf{u}(x,y) = \begin{bmatrix} \sin(x)\sin(y) \\ \cos(x)\cos(y) \end{bmatrix} + \frac{1}{\lambda}\begin{bmatrix} x \\ y \end{bmatrix}. \tag{51}$$

Here the first term is divergence-free, but the second term provides a constant divergence $(2/\lambda)$. Clearly, $\nabla \cdot \mathbf{u} \to 0$ as $\lambda \to \infty$. But $\mu = 0.5$ is fixed.

The results in Table 1 indicate that the Lagrangian Q_1^2 finite element method for linear elasticity on rectangular meshes is subject to Poisson-locking. When $\lambda = 1$, the errors in displacement, stress, and dilation exhibit 2nd, 1st, and 1st order convergence, respectively. But when $\lambda = 10^6$, there is no convergence in the displacement errors.

However, the results in Table 2 demonstrate that the new method based on the enriched Lagrangian elements (EQ_1) is locking-free. As λ is increased from 1 to 10^6, the convergence rates in displacement and stress (relative errors) are maintained as 2nd order and 1st order, respectively. The convergence rate in dilation errors is actually better than the theoretical estimate 1st order. This may be related to the fact that the exact dilation is a constant.

In addition, we have also tested other analytical examples in the literature, e.g., [33] (p.134 therein) and [14] (p.997 therein). Our numerical results demonstrate the locking-free property of our solver and its optimal 2nd order convergence in displacement, 1st order in stress, and 1st order in dilation. However, the details are omitted due to page limitation.

Example 2 (*Cook's Membrane*)**.** For this test case, the domain Ω is a trapezoid that connects the four vertices $(0,0)$, $(48,44)$, $(48,60)$, and $(0,44)$.

Table 1
Example 1: CG.Q_1^2 on rectangular meshes suffers Poisson-locking.

$1/h$	$\|\mathbf{u}-\mathbf{u}_h\|$	Rate	$\|\sigma-\sigma_h\|$	Rate	$\|\nabla\cdot\mathbf{u}-\nabla\cdot\mathbf{u}_h\|$	Rate
$\lambda=1$						
8	2.1169e−03	–	6.5673e−02	–	2.7993e−02	–
16	5.2969e−04	1.99	3.2859e−02	0.99	1.4010e−02	0.99
32	1.3244e−04	1.99	1.6432e−02	0.99	7.0066e−03	0.99
64	3.3111e−05	2.00	8.2166e−03	0.99	3.5035e−03	0.99
128	8.2777e−06	2.00	4.1083e−03	1.00	1.7518e−03	0.99
$\lambda=10^6$						
8	4.7193e−03	–	3.9423e+04	–	2.7876e−02	–
16	3.7879e−03	0.31	1.9728e+04	0.99	1.3949e−02	0.99
32	3.6288e−03	0.06	9.8659e+03	0.99	6.9763e−03	0.99
64	3.5810e−03	0.01	4.9332e+03	0.99	3.4883e−03	0.99
128	3.5145e−03	0.02	2.4667e+03	0.99	1.7442e−03	0.99

Table 2
Example 1: CG.EQ$_1$ on rectangular meshes is locking-free.

$1/h$	$\|\mathbf{u}-\mathbf{u}_h\|/\|\mathbf{f}\|$	Rate	$\|\sigma-\sigma_h\|/\|\mathbf{f}\|$	Rate	$\lambda\|\nabla\cdot\mathbf{u}-\overline{\nabla\cdot\mathbf{u}_h}\|/\|\mathbf{f}\|$	Rate
$\lambda=1$						
8	2.3044e−03	–	3.6811e−02	–	1.0862e−03	–
16	5.7523e−04	2.00	1.8212e−02	1.01	3.0293e−04	1.84
32	1.4377e−04	2.00	9.0616e−03	1.00	8.2803e−05	1.87
64	3.5945e−05	1.99	4.5204e−03	1.00	2.2325e−05	1.89
128	8.9875e−06	1.99	2.2577e−03	1.00	5.9596e−06	1.90
$\lambda=10^6$						
8	2.5437e−03	–	3.7236e−02	–	3.9230e−03	–
16	6.3700e−04	1.99	1.8283e−02	1.02	1.1226e−03	1.80
32	1.5943e−04	1.99	9.0727e−03	1.01	3.1295e−04	1.84
64	3.9886e−05	1.99	4.5220e−03	1.00	8.5603e−05	1.87
128	9.9807e−06	1.99	2.2579e−03	1.00	2.2944e−05	1.89

Table 3
Poisson's ratio (ν) values of commonly seen materials.

Material	ν	Material	ν
Concrete	0.1 – 0.2	Rubber	0.4999
Copper	0.33	Saturated clay	0.40 – 0.49
Foam	0.1 – 0.5	Stainless steel	0.30 – 0.31

The left side is clamped, namely, we have a displacement boundary condition $\mathbf{u}_D=[0,0]^T$ for $x=0$. The right side has a vertical traction with a total load 1, in other words, there is a traction boundary condition $\mathbf{t}_N=[0,\frac{1}{16}]^T$ pointwise for $x=48$. There is no body force, i.e., $\mathbf{f}=[0,0]^T$. The elasticity modulus is $E=1$ but the Poisson's ratio ν varies. There is no known analytical solution for this problem. This example has been frequently tested [5,34], see Fig. 3 for an illustration.

For this example, it is natural to use quadrilateral meshes that respect the domain geometry. We use both the classical Lagrangian elements Q_1^2 and the enriched elements EQ$_1$ in the continuous Galerkin framework. After a numerical displacement is obtained, the numerical dilation and stress are calculated *ad hoc*. To compare the performances of these two finite element solvers, we choose ν as 0.33, 0.49, or 0.4999. As shown in Table 3, these mimic elastic materials of copper, saturated clay, and rubber, respectively.

Figs. 4 and 5 present profiles of numerical displacement, dilation, and shear stress σ_{xy} obtained from using CG.Q_1^2 and CG.EQ$_1$ on a 32×32 quadrilateral mesh. It is obvious that for CG.Q_1^2, all these three quantities start to behave suspiciously (see the right column of each figure), as ν approaches 0.5, or the material becomes nearly incompressible. However, for CG.EQ$_1$, all these three quantities are reasonably correct (see the left column of each figure), as reflected by the following observations: Regardless of the Poisson's ratio (ν value) of the material, due to the mechanical conditions (displacement and traction data, and no body force),

(i) Around the top-left corner, the elastic body is squeezed the most, i.e., the dilation $\nabla\cdot\mathbf{u}$ is negative and has the largest magnitude;
(ii) The body is stretched the most near the bottom side, that is, $\nabla\cdot\mathbf{u}>0$ with the largest magnitude.

Example 3 (*A Cantilever Beam*)**.** For this frequently tested example, we consider a rectangular beam that has length a, height b, and unit thickness. The beam is subject to a couple at one end and a linearly varying load with magnitude F at

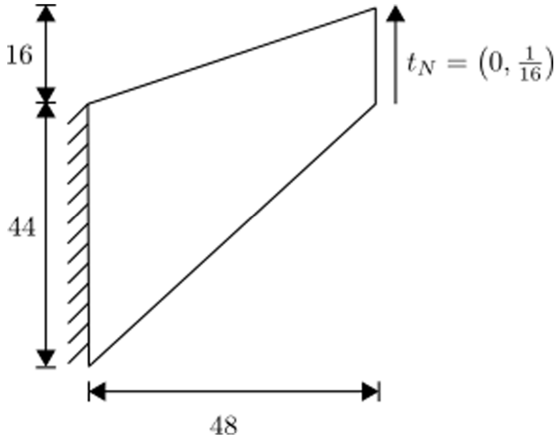


Fig. 3. Example 2 (Cook's membrane): Problem illustration.

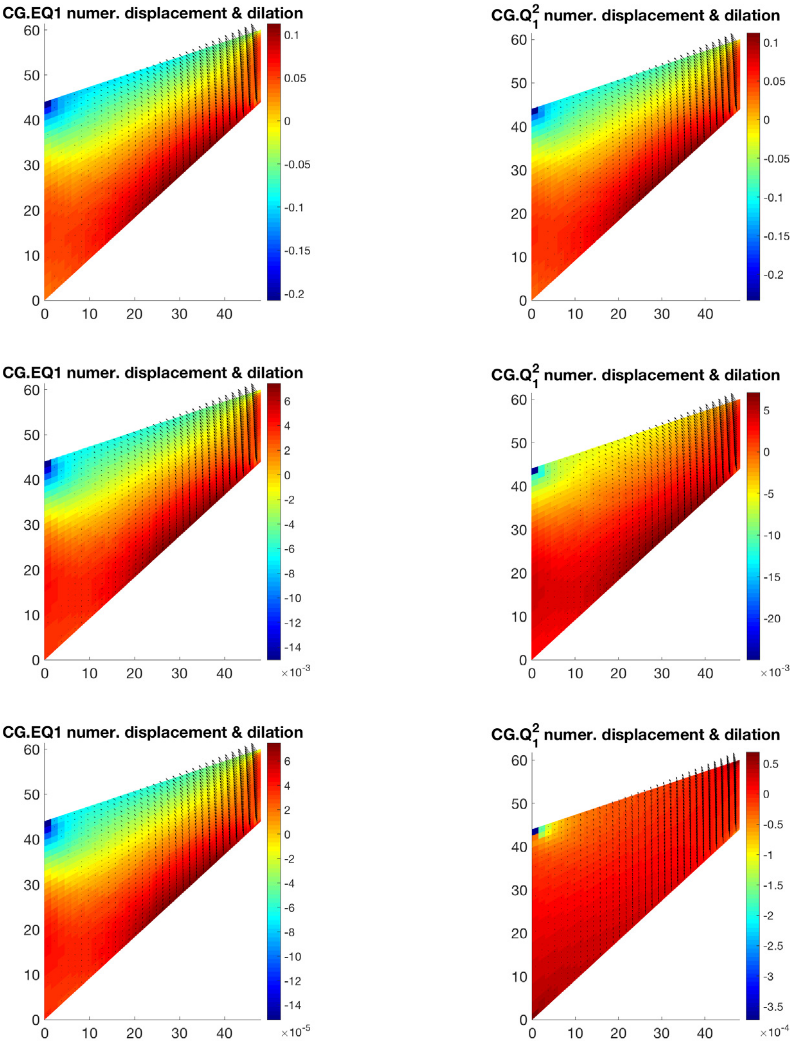


Fig. 4. Example 2 (Cook's membrane): Numerical displacement and dilation obtained from using CG.EQ$_1$ (*left column*) and CG.Q_1^2 (*right column*) on a 32×32 quadrilateral mesh. *Top row:* $\nu = 0.33$ (copper); *Middle row:* $\nu = 0.49$ (saturated clay); *Bottom row:* $\nu = 0.4999$ (rubber). As ν approaches 0.5, CG.Q_1^2 results become suspicious but CG.EQ$_1$ results are correct.

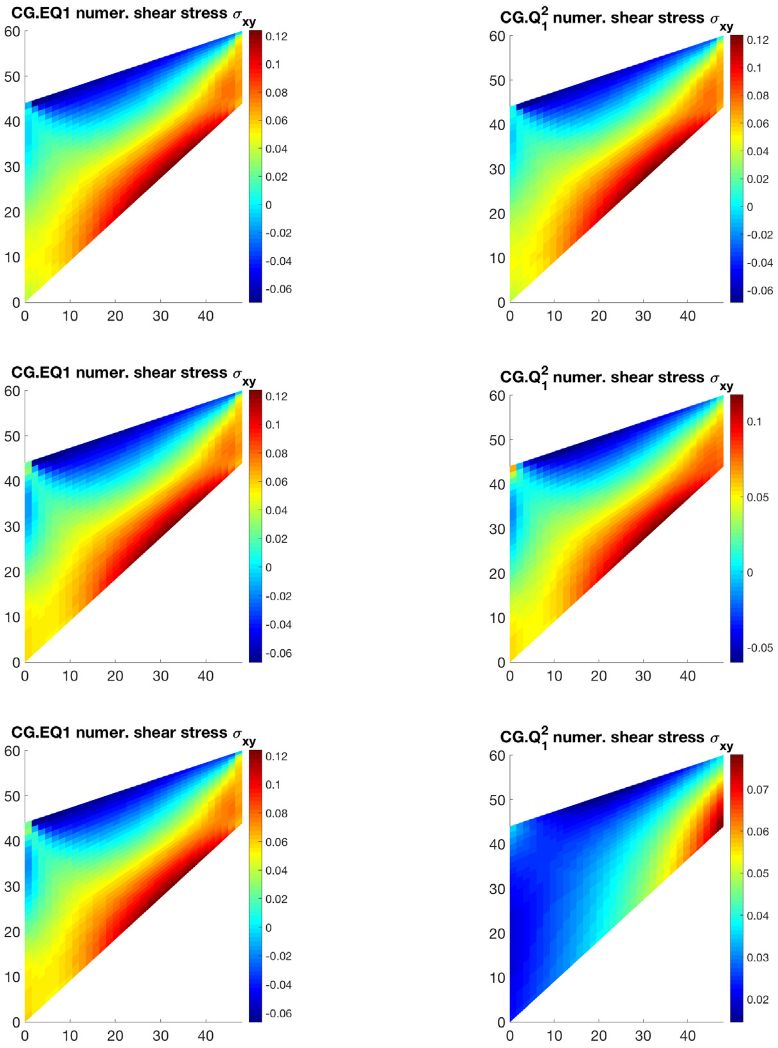


Fig. 5. Example 2 (Cook's membrane): Numerical shear stress σ_h^{xy} obtained from using CG.EQ$_1$ (*left column*) and CG.Q_1^2 (*right column*) on a 32 × 32 quadrilateral mesh. *Top row:* $\nu = 0.33$; *Middle row:* $\nu = 0.49$; *Bottom row:* $\nu = 0.4999$. As ν approaches 0.5, CG.Q_1^2 results become suspicious but CG.EQ$_1$ results are correct.

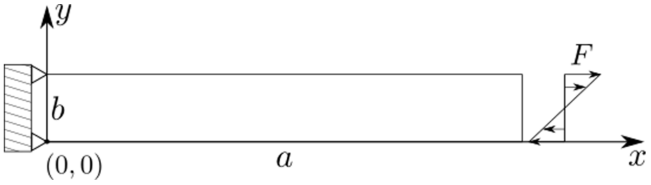


Fig. 6. Example 3 (A cantilever beam): A beam with length a and height b is coupled at the left end and subject to a linearly varying load at the right end.

the free end, as shown in Fig. 6. Along the edge $x = 0$, the horizontal displacement is zero. At the lower-left corner (also the origin) and the upper-left corner, the vertical displacement is also zero. In other words,

$$u_1|_{x=0}= 0, \qquad \mathbf{u}|_{(0,0)}= \mathbf{u}|_{(0,b)}= \mathbf{0}, \qquad \mathbf{t}_N|_{x=a}= \begin{bmatrix} \dfrac{2y-b}{b}F \\ 0 \end{bmatrix}. \tag{52}$$

Table 4
Example 3: A cantilever beam solved by CG.EQ$_1$: Errors and convergence rates for $(10n)\times n$ rectangular meshes.

n	$\|\mathbf{u}-\mathbf{u}_h\|$	Rate	$\|\sigma-\sigma_h\|$	Rate	$\max(\sigma_h^{xx})$
$\nu = 0.4$					
4	1.5241e−02	–	9.0535e−01	–	3.8519
8	3.7399e−03	2.02	4.5553e−01	0.99	4.4211
16	9.3084e−04	2.00	2.2810e−01	0.99	4.7105
32	2.3247e−04	2.00	1.1409e−01	0.99	4.8552
64	5.8093e−05	2.00	5.7053e−02	0.99	4.9276
$\nu = 0.49999$					
4	1.7020e−2	–	1.1392e+00	–	3.8787
8	4.1565e−3	2.03	5.7035e−01	0.99	4.4336
16	1.0333e−3	2.00	2.8524e−01	0.99	4.7167
32	2.5787e−4	2.00	1.4263e−01	0.99	4.8583
64	6.3857e−5	2.01	7.1316e−02	0.99	4.9292

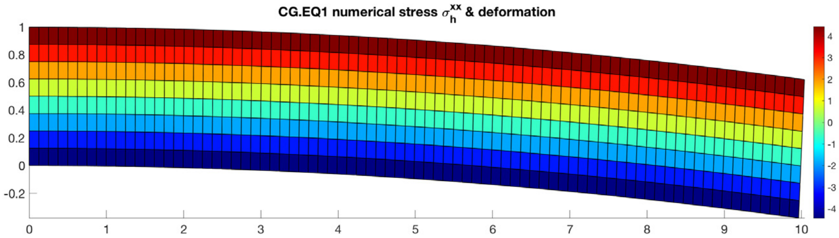


Fig. 7. Example 3 (A cantilever beam) with $\nu = 0.499$: Numerical normal stress σ_h^{xx} and deformation obtained from applying CG.EQ$_1$ on a 80×8 rectangular mesh.

We ignore the beam's own weight, so there is no body force exerting on the beam, i.e., $\mathbf{f}=\mathbf{0}$. It is known [5,12] that the exact solution for displacement is $\mathbf{u}=[u_1,u_2]^T$ with

$$u_1(x,y) = Ax(2y-b), \qquad u_2(x,y) = -Ax^2 + By(b-y), \tag{53}$$

where

$$A = \frac{1-\nu^2}{E}\frac{F}{b}, \qquad B = \frac{\nu}{1-\nu}A. \tag{54}$$

Obviously,

$$\nabla\cdot\mathbf{u} = \frac{(1-2\nu)(1+\nu)}{E}\frac{F}{b}(2y-b). \tag{55}$$

So the upper half of the beam $(y > b/2)$ is stretched, whereas the lower half $(y < b/2)$ is compressed. Moreover, $\nabla\cdot\mathbf{u}\to 0$ as $\nu\to\frac{1}{2}$. It is also clear by direct calculations that

$$\sigma_{xx} = (2\mu A + \lambda(A-B))(2y-b) = \frac{F}{b}(2y-b), \quad \sigma_{yy}=\sigma_{xy}=\sigma_{yx}=0. \tag{56}$$

Hence the maximum of the normal stress in the x-direction is $\max(\sigma_{xx}) = F$.

An interesting phenomenon in this example is that many quantities, e.g., displacement $\mathbf{u}$ and stress σ, are actually independent of the beam length a.

Note also that our description for the cantilever beam is slightly different than what is commonly seen in the literature (up to a negative sign) but looks more reasonable for simulating a bending-down beam.

For numerical experiments, we set $a = 10$, $b = 1$, $E = 1000$, $F = 5$ but vary ν among 0.4, 0.499, and 0.49999. We use a family of uniform rectangular meshes with $(10n)\times n$ elements, where $n = 4, 8, 16, 32, 64$, respectively. Shown in Table 4 are the errors in displacement, stress, and dilation obtained from applying the CG.EQ$_1$ method. The convergence rates are respectively 2nd, 1st, and 1st order, as expected, regardless of the value of the Poisson's ratio ν. The maximum of the numerical normal stress $\max(\sigma_h^{xx})$ approaches the analytical value 5.0, and the convergence rate is also 1st order.

Shown in Fig. 7 are the profiles of the numerical normal stress σ_h^{xx} and deformation (determined by displacement nodal values) obtained from applying the EQ$_1$ solver developed in this paper on a 80×8 rectangular mesh. In this case, $\nu = 0.499$, one can clearly observe the symmetry of the numerical stress σ_h^{xx} across the middle line $y = \frac{1}{2}$ and the downward bending in the beam, due to the linearly varying load on the right (free) end.

Table 5
Example 4: CG.EQ$_1$ convergence rates for slightly distorted hexahedral meshes.

$1/h$	$\|\mathbf{u}-\mathbf{u}_h\|$	Rate	$\|\sigma-\sigma_h\|/\|\mathbf{f}\|$	Rate	$\lambda\|\nabla\cdot\mathbf{u}-\overline{\nabla\cdot\mathbf{u}_h}\|/\|\mathbf{f}\|$	Rate
$\lambda=1$						
4	8.7462e−02	–	1.1379e−01	–	3.3496e−02	–
8	2.3135e−02	1.91	5.5831e−02	1.02	1.7069e−02	0.97
16	5.9074e−03	1.96	2.7713e−02	1.01	8.5971e−03	0.98
32	1.4671e−03	2.00	1.3788e−02	1.00	4.3071e−03	0.99
$\lambda=10^3$						
4	1.7364e−02	–	1.9304e−01	–	1.1127e−01	–
8	4.5972e−03	1.91	9.9476e−02	0.95	5.6893e−02	0.96
16	1.1970e−03	1.94	5.0429e−02	0.98	2.8514e−02	0.99
32	3.0264e−04	1.98	2.5346e−02	0.99	1.4264e−02	0.99

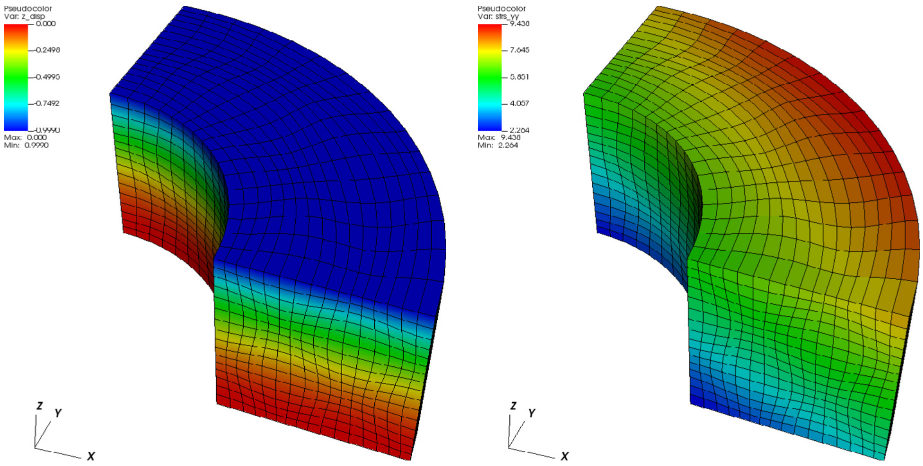


Fig. 8. Example 4 (Slightly distorted hexahedral meshes): Elasticity of a cylindrical sector ($\lambda=10^3$) solved by CG.EQ$_1$. *Left panel*: Profile of numerical displacement z-component (with mesh) for $h=1/16$; *Right panel*: Profile of numerical normal stress σ_h^{yy} (with mesh) for $h=1/16$.

Example 4 (*Elasticity on a Cylindrical Domain*)**.** Here we consider a cylindrical sector described by

$$\Omega=\{(r,\theta,z): r\in(1,2), \theta\in(0,\pi/2), z\in(0,1)\}.$$

An analytical solution for displacement is specified as

$$\mathbf{u}=A\begin{bmatrix}xz\\ yz\\ -z^2\end{bmatrix}+\frac{B}{\lambda}\begin{bmatrix}x^2\\ y^2\\ z^2\end{bmatrix},$$

where $A>0$, $B>0$ are constants. The dilation is $\nabla\cdot\mathbf{u}=2(x+y+z)B/\lambda$ so that $\nabla\cdot\mathbf{u}\to 0$ as $\lambda\to\infty$. The stress and body force are derived accordingly. A displacement boundary condition is posed on all boundary faces.

For numerical experiments, we fix $A=1$, $B=1$, $\mu=1$, but vary λ to test the locking-free property of this new solver. The numerical results for this example were produced using our `deal.II` code. We have tested both uniform hexahedral meshes (similar to those used in [32]) and slightly distorted hexahedral meshes. For the latter, we use smooth perturbations of the cylindrical coordinates of the former. Specifically, if the coordinates of the nodes in a uniform mesh are $(\hat{r},\hat{\theta},\hat{z})$, then the coordinates of the nodes in the slightly distorted mesh are given as

$$\begin{cases} r=\hat{r}+0.02\sin(3\pi(\hat{r}-1))\cos(6\hat{\theta})\cos(3\pi\hat{z}),\\ \theta=\hat{\theta}+0.02\cos(3\pi(\hat{r}-1))\sin(6\hat{\theta})\cos(3\pi\hat{z}),\\ z=\hat{z}+0.02\cos(3\pi(\hat{r}-1))\cos(6\hat{\theta})\sin(3\pi\hat{z}).\end{cases} \tag{57}$$

Convincing numerical results have been obtained for both types of meshes. Due to page limitation, we focus on the latter. As shown in Table 5, the absolute or relative errors in displacement, stress, and dilation exhibit 2nd, 1st, and 1st order convergence, respectively, regardless of the values of λ. In Fig. 8 generated with the visualization software `VisIt` [35], the profile of numerical normal stress σ_h^{yy} exhibits linear variation as expected.

6. Concluding remarks

In this paper, we have developed a new finite element solver for linear elasticity on quadrilateral and hexahedral meshes based on enrichment of the classical Lagrangian Q_1^2 or Q_1^3 elements by edge/face-based bubble functions. This new solver is locking-free, as validated by the theoretical analysis in Section 3 and the numerical experiments on benchmarks, e.g., Cook's membrane and a cantilever beam, in Section 5.

This new solver has been implemented in different forms: (i) In Matlab for two-dimensional elasticity problems, the code modules are included in our own package DarcyLite; (ii) In C++ for both two-dimensional and three-dimensional elasticity problems in the so-called dimension-independent paradigm, the code modules are included in the popular FEM package deal.II. We intend to make this new solver freely accessible to the scientific computing community.

Different than the mixed, HDG, IPDG finite element methods for linear elasticity, our new method is a CG type method. It is a locking-free method in the primal formulation, resulting in a symmetric positive-definite linear system that is easy to solve. However, the recovered numerical stress does not have the strong continuity that is built in the construction of some mixed methods.

As is well known, the classical Lagrangian Q_1^2 or Q_1^3 elements suffer Poisson-locking when used for solving elasticity problems [1,2]. The results in this paper have demonstrated that we obtain a new locking-free elasticity solver by appropriately enriching those classical $Q_1^d(d = 2, 3)$ spaces. This echoes our early efforts in [36]. It is known that the Lagrangian P_1 or Q_1 elements are not locally mass-conservative when used for solving the Darcy equation. However, in [36], we have demonstrated theoretically and numerically that we obtain new locally mass-conservative Darcy solvers by simply enriching the classical P_1 or Q_1 spaces with elementwise constants.

In the bilinear form defined in (18), the treatment of the dilation term may be interpreted as *reduced integration*, as done in [3]. Such techniques date back to [37]. Here it is actually more convenient to understand it as a mimicry of matching the EQ_1 or BR_1 space (for displacement) to the constant space Q_0 (for the pseudo-pressure).

Our numerical solver is motivated by the Stokes element pair (BR_1, Q_0) on quadrilaterals [28]. But we are not directly using the (BR_1, Q_0) element pair. We are solving for only the displacement and the scheme (17) is in the primal formulation.

The two-dimensional version of this new elasticity solver has also been combined with a two-dimensional Darcy solver based on the novel weak Galerkin methodology for development of a new poroelasticity solver [38], which was implemented in Matlab. The promising numerical results in [38] have demonstrated potentials for reviving the 2-field approach for poroelasticity, which calls for further development and analysis of efficient solvers. This is currently under our investigation and will be reported in our future work.

Extension to higher order is possible. This shall involve bubble functions similar to those used in [23] and reduced integration techniques [37]. Implementation of these higher order solvers on open-access platforms are undoubtedly useful. These will be our future work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

G. Harper and J. Liu were partially supported by US National Science Foundation under grant DMS-1819252. S. Tavener was partially supported by US National Science Foundation under grant DMS-1720473/1720402. R. Wang was partially supported by China Postdoctoral Science Foundation through grant 2019M661199 and Postdoctoral Innovative Talent Support Program (BX20190142). R. Zhang was supported in part by National Natural Science Foundation of China (grant 11971198, 11871245, 11771179, 11826101) and the Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education of China (housed at Jilin University). We sincerely thank the anonymous reviewers for their insightful comments, which have helped improve the quality of this paper.

References

[1] S. Brenner, L. Scott, The Mathematical Theory of Finite Element Methods, third ed., in: Texts in Applied Mathematics, vol. 15, Springer-Verlag, New York, 2008.

[2] P.J. Phillips, M.F. Wheeler, Overcoming the problem of locking in linear elasticity and poroelasticity: an heuristic approach, Comput. Geosci. 13 (2009) 5–12.

[3] S. Brenner, L.-Y. Sung, Linear finite element methods for planar linear elasticity, Math. Comp. 59 (1992) 321–338.

[4] J. Hu, H. Man, J. Wang, S. Zhang, The simplest nonconforming mixed finite element method for linear elasticity in the symmetric formulation on n-rectangular grids, Comput. Math. Appl. 71 (2016) 1317–1336.

[5] B.P. Lamichhane, A mixed finite element method for nearly incompressible elasticity and Stokes equations using primal and dual meshes with quadrilateral and hexahedral grids, J. Comput. Appl. Math. 260 (2014) 356–363.

[6] D. Mijuca, On hexahedral finite element HC8/27 in elasticity, Comput. Mech. 33 (2004) 466–480.

[7] D. Arnold, R. Winther, Mixed finite elements for elasticity, Numer. Math. 92 (2002) 401–419.

[8] D. Arnold, G. Awanou, Rectangular mixed finite elements for elasticity, Math. Model. Methods Appl. Sci. 15 (2005) 1417–1429.

[9] D. Arnold, G. Awanou, W. Qiu, Mixed finite elements for elasticity on quadrilateral meshes, Adv. Comput. Math. 41 (2015) 553–572.

[10] J. Hu, H. Man, S. Zhang, A simple conforming mixed finite element for linear elasticity on rectangular grids in any space dimension, J. Sci. Comput. 58 (2014) 367–379.

[11] G. Awanou, A rotated nonconforming rectangular mixed element for elasticity, Calcolo 46 (2009) 49–60.

[12] A. Chama, B. Reddy, New stable mixed finite element approximations for problems in linear elasticity, Comput. Methods Appl. Mech. Engrg. 256 (2013) 211–223.

[13] S. Mao, S. Chen, A quadrilateral nonconforming finite element for linear elasticity problem, Adv. Comput. Math. 28 (2008) 81–100.

[14] S.-Y. Yi, A new nonconforming mixed finite element method for linear elasticity, Math. Model. Methods Appl. Sci. 16 (2006) 979–999.

[15] B. Riviére, S. Shaw, M.F. Wheeler, J.R. Whiteman, Discontinuous Galerkin finite element methods for linear elasticity and quasistatic linear viscoelasticity, Numer. Math. 95 (2003) 347–376.

[16] B. Riviére, M.F. Wheeler, Optimal Error Estimates for Discontinuous Galerkin Metjhods Applied to Linear Elasticity Problems, Technical report, Texas Institute for Computational and Applied Mathematics, 1999.

[17] P. Hansbo, M.G. Larson, Discontinuous Galerkin methods for incompressible and nearly incompressible elasticity by Nitsche's method, Comput. Methods Appl. Mech. Engrg. 191 (2002) 1895–1908.

[18] S.-C. Soon, B. Cockburn, H.K. Stolarski, A hybridizable discontinuous Galerkin method for linear elasticity, Internat. J. Numer. Methods Engrg. 80 (2009) 1058–1092.

[19] C. Wang, J. Wang, R. Wang, R. Zhang, A locking-free weak Galerkin finite element method for elasticity problems in the primal formulation, J. Comput. Appl. Math. 307 (2016) 346–366.

[20] G. Harper, J. Liu, S. Tavener, B. Zheng, Lowest-order weak Galerkin finite element methods for linear elasticity on rectangular and brick meshes, J. Sci. Comput. 78 (3) (2019) 1917–1941.

[21] S.-Y. Yi, A lowest-order weak Galerkin method for linear elasticity, J. Comput. Appl. Math. 350 (2019) 286–298.

[22] B.P. Lamichhane, E.P. Stephan, A symmetric mixed finite element method for nearly incompressible elasticity based on biorthogonal systems, Numer. Methods Partial Differential Equations 28 (2012) 1336–1353.

[23] S.-Y. Yi, A study of two modes of locking in poroelasticity, SIAM J. Numer. Anal. 55 (2017) 1915–1936.

[24] D.N. Arnold, D. Boffi, R.S. Falk, Approximation by quadrilateral finite elements, Math. Comp. 71 (2002) 909–922.

[25] R.S. Falk, P. Gatto, P. Monk, Hexahedral H(div) and H(curl) finite elements, Math. Model. Numer. Anal. 45 (2011) 115–143.

[26] J. Liu, S. Tavener, Z. Wang, Penalty-free any-order weak Galerkin FEMs for elliptic problems on quadrilateral meshes, J. Sci. Comput. 83 (2020) 47.

[27] Y. Zhang, S. Wang, D. Chan, A new five-node locking-free quadrilateral element based on smoothed FEM for near-incompressible linear elasticity, Internat. J. Numer. Methods Engrg. 100 (2014) 633–668.

[28] C. Bernardi, G. Raugel, Analysis of some finite elements for the Stokes problem, Math. Comp. 44 (1985) 71–79.

[29] D. Arndt, W. Bangerth, T.C. Clevenger, D. Davydov, M. Fehling, D. Garcia-Sanchez, G. Harper, T. Heister, L. Heltai, M. Kronbichler, R.M. Kynch, M. Maier, J.-P. Pelteret, B. Turcksin, D. Wells, The deal.II library, version 9.1, J. Numer. Math. 27 (2019).

[30] S. Brenner, Korn's inequalities for piecewise H1 vector fields, Math. Comp. 73 (2004) 1067–1087.

[31] M. Crouzeix, P. Raviart, Conforming and nonconforming finite element methods for solving the stationary Stokes equations I, RAIRO 7 (1973) 33–75.

[32] G. Harper, J. Liu, B. Zheng, The THex algorithm and a simple Darcy solver on hexahedral meshes, Procedia Comput. Sci. 108C (2017) 1903–1912.

[33] S. Brenner, A nonconforming mixed multigrid method for the pure displacement problem in planar linear elasticity, SIAM J. Numer. Anal. 30 (1993) 116–135.

[34] C. Carstensen, M. Schedensack, Medius analysis and comparison results for first-order finite element methods in linear elasticity, IMA J. Numer. Anal. 35 (2015) 1591–1621.

[35] H. Childs, E. Brugger, B. Whitlock, J. Meredith, S. Ahern, D. Pugmire, K. Biagas, M. Miller, C. Harrison, G.H. Weber, H. Krishnan, T. Fogal, A. Sanderson, C. Garth, E.W. Bethel, D. Camp, O. Rübel, M. Durant, J.M. Favre, P. Navrátil, VisIt: An end-user tool for visualizing and analyzing very large data, in: High Performance Visualization–Enabling Extreme-Scale Scientific Insight, 2012, pp. 357–372.

[36] S. Sun, J. Liu, A locally conservative finite element method based on piecewise constant enrichment of the continuous Galerkin method, SIAM J. Sci. Comput. 31 (2009) 2528–2548.

[37] D.S. Malkus, T. Hughes, Mixed finite element methods - reduced and selective integration techniques: A unification of concepts, Comput. Methods Appl. Mech. Engrg. 15 (1978) 63–81.

[38] G. Harper, J. Liu, S. Tavener, Z. Wang, A two-field finite element solver for poroelasticity on quadrilateral meshes, Lect. Notes Comput. Sci. 10862 (2018) 76–88.