



ARTICLE

Matrix-Free Higher-Order Finite Element Method for Parallel Simulation of Compressible and Nearly-Incompressible Linear Elasticity on Unstructured Meshes

Dedicated to Professor Karl Stark Pister for his 95th birthday

Arash Mehraban¹, Henry Tufo¹, Stein Sture² and Richard Regueiro^{2,*}

¹Department of Computer Science, University of Colorado Boulder, Boulder, CO, USA

²Department of Civil, Environmental, and Architectural Engineering, University of Colorado Boulder, Boulder, CO, USA

*Corresponding Author: Richard Regueiro. Email: richard.regueiro@colorado.edu

Received: 13 May 2021 Accepted: 02 July 2021

ABSTRACT

Higher-order displacement-based finite element methods are useful for simulating bending problems and potentially addressing mesh-locking associated with nearly-incompressible elasticity, yet are computationally expensive. To address the computational expense, the paper presents a matrix-free, displacement-based, higher-order, hexahedral finite element implementation of compressible and nearly-compressible ($\nu \rightarrow 0.5$) linear isotropic elasticity at small strain with p -multigrid preconditioning. The cost, solve time, and scalability of the implementation with respect to strain energy error are investigated for polynomial order $p = 1, 2, 3, 4$ for compressible elasticity, and $p = 2, 3, 4$ for nearly-incompressible elasticity, on different number of CPU cores for a tube bending problem. In the context of this matrix-free implementation, higher-order polynomials ($p = 3, 4$) generally are faster in achieving better accuracy in the solution than lower-order polynomials ($p = 1, 2$). However, for a beam bending simulation with stress concentration (singularity), it is demonstrated that higher-order finite elements do not improve the spatial order of convergence, even though accuracy is improved.

KEYWORDS

Matrix-free; higher-order; finite element; parallel; linear elasticity; multigrid solvers; unstructured meshes

1 Introduction

Modeling bending deformation and associated stress of compressible and nearly-incompressible linear isotropic elastic materials via the standard linear interpolation displacement-based Finite Element Method (FEM) may lead to inaccurate stress calculations (depending on element size h) as well as potential mesh-locking [1,2]. With respect to accuracy, either h -or p -refinement are adopted, but oftentimes not the two together because of the associated high computational cost. With respect to mesh-locking, several FEM approaches involving mixed formulations have been proposed to overcome volumetric strain locking of linear finite elements [3–6]. Most such mixed methods for linear elasticity at small strain solve for two fields (displacement and pressure), which



are discretized separately via an additive split of the deviatoric and volumetric parts of the small strain tensor [7,8]. As an alternative to, and oftentimes in conjunction with, mixed formulations, higher-order finite elements have been adopted to attempt to overcome mesh locking [9]. However, assembling a global Jacobian matrix based on higher-order finite element interpolation functions leads to expensive solve times as a result of higher memory requirements [10]. Matrix-free formulations [11,12] provide the benefits of higher-order finite element methods but with more efficient memory usage. Concurrently, using tensor-product-basis evaluation further improves the performance of higher-order methods by supplying favorable storage and work estimates of $O(p^d)$ and $O(p^{d+1})$ per p -order element, respectively, for discretizations in \mathbb{R}^d [11], where d denotes spatial dimension. Furthermore, tensor-product-based operator evaluations can be cast as matrix-matrix products. Iterative solvers, such as Krylov subspace methods, only require the result of matrix-vector products rather than an assembled matrix, which allows these methods to take advantage of the computational efficiencies offered by matrix-free formulations. Therefore, storage of large, sparse matrices is avoided with iterative solvers.

Krylov subspace methods require preconditioning [13] to efficiently and stably solve large-scale elliptic problems such as 3D linear isotropic elasticity within the static balance of linear momentum equations. Geometric multigrid is a robust preconditioning scheme that is an appealing choice for structured meshes, and has been considered in numerous studies using finite difference and finite element methods with tensor-product-bases implemented on CPUs and GPUs [14–16]. p -multigrid, developed by Rønquist et al. [17], is a version of geometric multigrid based on coarsening by decreasing the basis order in higher-order or spectral finite elements rather than coarsening by aggregating elements. As a result, p -multigrid is a natural approach for problems on unstructured meshes, whereby spatial convergence only weakly depends on polynomial order p , if properly implemented. Computational cost of the discretization scheme may be affected by polynomial order p and element size h , among other factors [18,19], such as adaptive r -refinement of the mesh [20].

In terms of engineering applications, compressible linear isotropic elasticity (e.g., $\nu = 0.3$) is appropriate for modeling metals in their elastic regime, such as in serviceability studies to analyze the stress intensity factor at a crack tip to determine if the crack will propagate and thus estimate the life cycle of a metallic component. Nearly-incompressible linear isotropic elasticity ($\nu \rightarrow 0.5$) is appropriate for modeling small deformations (small strains and small rotations) of rubber-like materials such as solid rock propellant binders [21]. When metals or rubber-like materials are loaded beyond their small strain linear elastic limit, then large deformation hyperelasticity or hyper-elasto-plasticity constitutive models are needed, which are beyond the scope of this paper.

In this work, we apply a previously-developed matrix-free, higher-order, FEM for solving three-dimensional (3D) linear isotropic elasticity with p -multigrid preconditioning [22,23] to (i) a tube geometry subjected to method of manufactured solutions (MMS), (ii) a tube bending problem for compressible ($\nu = 0.3$) and nearly-incompressible ($\nu = 0.499999$) elasticity and polynomial order $p = 1, 2, 3, 4$, and (iii) a beam bending problem under body force loading for which the beam is clamped on both ends. Simulations (ii) and (iii)—but simulation (iii) in particular—cause a stress concentration (singularity) in the clamped regions, where the higher-order FEM may not converge optimally as expected [22]. Therefore, we investigate order of convergence of the matrix-free higher-order FEM for compressible elasticity when there is a stress concentration. We aim to answer the following questions: (1) *With the matrix-free parallel implementation, what combination*

of mesh-refinement h and polynomial order p leads to a certain strain energy error that requires the least computational cost for compressible and nearly-incompressible elasticity using the standard displacement-based FEM? (2) What combination of mesh-refinement h and polynomial order p achieves a certain strain energy error for least amount of simulation time? (3) Does the higher-order, displacement-based FEM remain competitive as Poisson's ratio approaches the nearly-incompressible limit ($\nu \rightarrow 0.5$)? (4) Does the higher-order FEM converge as expected when there is a stress concentration present? The parallel computational toolkit PETSc [24] is employed for the linear solver as well as Algebraic MultiGrid (AMG) preconditioner for the coarse-grid solve, while libCEED [25] is used to perform efficient tensor-product-basis evaluation.

The rest of the paper is organized as follows. In Section 2, the constitutive model for compressible 3D linear isotropic elasticity and the variational form of static balance of linear momentum are presented. In Section 3, a preconditioning scheme used to accelerate convergence is briefly discussed. In Section 4, numerical results are presented, and in Section 5, observations and conclusions are provided.

2 Compressible Linear Isotropic Elasticity

For linear isotropic elastic materials, we consider the stored strain energy function as,

$$\Phi(\boldsymbol{\epsilon}) = \frac{\lambda}{2} \text{tr}(\boldsymbol{\epsilon})^2 + \mu \boldsymbol{\epsilon} : \boldsymbol{\epsilon} \quad (1)$$

Therefore, its stress-strain relationship is given by its derivative with respect to strain as,

$$\boldsymbol{\sigma} = \frac{\partial \Phi(\boldsymbol{\epsilon})}{\partial \boldsymbol{\epsilon}} = \lambda \text{tr}(\boldsymbol{\epsilon}) \mathbf{I} + 2\mu \boldsymbol{\epsilon}, \quad (2)$$

where $\boldsymbol{\sigma}$ and $\boldsymbol{\epsilon}$ are stress and strain tensors, respectively; and λ and μ are the Lamé parameters. The small strain tensor is,

$$\boldsymbol{\epsilon} = \frac{1}{2} (\nabla \mathbf{u} + (\nabla \mathbf{u})^T). \quad (3)$$

The strong form of the static balance of linear momentum is given by the following,

$$\begin{cases} \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g} = \mathbf{0} \\ \mathbf{u} = \bar{\mathbf{u}} \\ \boldsymbol{\sigma} \cdot \mathbf{n} = \bar{\mathbf{t}} \end{cases} \quad (4)$$

where the functions $\bar{\mathbf{u}}$ and $\bar{\mathbf{t}}$ are prescribed boundary displacement and traction, respectively, \mathbf{g} is the body force per unit mass (such as the gravitational acceleration vector), and ρ is the mass density. Introducing weighting function \mathbf{w} for the displacement field, the corresponding variational equation for (4) is written as,

$$\int_{\Omega} \nabla \mathbf{w} : \boldsymbol{\sigma} \, dv - \int_{\partial \Omega^{\bar{\mathbf{t}}}} \mathbf{w} \cdot \bar{\mathbf{t}} \, da - \int_{\Omega} \mathbf{w} \cdot \rho \mathbf{g} \, dv = 0. \quad (5)$$

2.1 Residual Evaluation

Discretization of the variational Eq. (5) can be written to facilitate matrix-free evaluation [10]. The residual in discrete form can be expressed as,

$$\sum_e \mathcal{E}_e^T \left[(\mathbf{N}^e)^T \cdot \mathbf{W} \cdot \Lambda(f_0(\mathbf{u}^e, \nabla \mathbf{u}^e)) + \sum_{i=1}^{\dim} (\mathbf{B}_i^e)^T \cdot \mathbf{W} \cdot \Lambda(f_1(\mathbf{u}^e, \nabla \mathbf{u}^e)) \right] = \mathbf{0} \quad (6)$$

where \mathbf{N}^e and \mathbf{B}_i^e are evaluations of the finite element shape functions and their derivatives at the quadrature points in x , y , and z directions, \mathcal{E}_e is the element e restriction operator that separates Degrees of Freedom (DoF) based on the elements to which they belong, and represents pointwise function evaluation. f_0 and f_1 are determined from the constitutive model and its tangent, where $\mathbf{u}^e = \mathbf{N}^e(\mathcal{E}_e \mathbf{d})$ and $\nabla \mathbf{u}^e = \sum_{i=1}^{\dim} [\mathbf{B}_i^e(\mathcal{E}_e \mathbf{d})]$, and \mathbf{d} is the total nodal displacement vector. The basis operators are represented as Kronecker products,

$$\begin{aligned} \mathbf{N}^e &= \hat{\mathbf{N}} \otimes \hat{\mathbf{N}} \otimes \hat{\mathbf{N}} & \mathbf{B}_1^e &= \hat{\mathbf{B}} \otimes \hat{\mathbf{N}} \otimes \hat{\mathbf{N}} \\ \mathbf{B}_2^e &= \hat{\mathbf{N}} \otimes \hat{\mathbf{B}} \otimes \hat{\mathbf{N}} & \mathbf{B}_3^e &= \hat{\mathbf{N}} \otimes \hat{\mathbf{N}} \otimes \hat{\mathbf{B}} \end{aligned} \quad (7)$$

where $\hat{\mathbf{N}}$ and $\hat{\mathbf{B}}$ evaluations of the finite element shape functions and their derivatives at the quadrature points in 1D. Comparing the variational equation in (5) to its discrete form in (6) yields $f_0 = \mathbf{g}$ and $f_1 = \boldsymbol{\sigma}$, assuming $\bar{\mathbf{t}} = \mathbf{0}$.

2.2 Jacobian Evaluation

Similar to a residual evaluation, the action of the Jacobian can be computed using the notation proposed by [10,26], as,

$$\frac{\partial \mathbf{F}(\mathbf{u})}{\partial \mathbf{u}} = \sum_e \mathcal{E}_e^T \left[\mathbf{N}^T \mathbf{B}^T \right] \mathbf{W} \begin{bmatrix} f_{0,0} & f_{0,1} \\ f_{1,0} & f_{1,1} \end{bmatrix} \begin{bmatrix} \mathbf{N} \\ \mathbf{B} \end{bmatrix} \mathcal{E}_e \quad (8)$$

where

$$\mathbf{f}_{i,j} = \begin{bmatrix} \frac{\partial f_0}{\partial \mathbf{u}} & \frac{\partial f_0}{\partial \nabla \mathbf{u}} \\ \frac{\partial f_1}{\partial \mathbf{u}} & \frac{\partial f_1}{\partial \nabla \mathbf{u}} \end{bmatrix} (\mathbf{u}, \nabla \mathbf{u}) \quad (9)$$

In the small strain case, for Eq. (2), f_0 is not a function of \mathbf{u} or $\nabla \mathbf{u}$. Therefore, its derivative with respect to \mathbf{u} or $\nabla \mathbf{u}$ is zero, (i.e., $f_{0,0} = \mathbf{0}$ and $f_{0,1} = \mathbf{0}$). On the other hand, f_1 is a function of $\nabla \mathbf{u}$, but it is not a function of \mathbf{u} . Therefore, $f_{1,0} = \mathbf{0}$ and $f_{1,1} = \partial \boldsymbol{\sigma} / \partial \boldsymbol{\epsilon}$; due to linearity, $\partial \boldsymbol{\sigma} / \partial \boldsymbol{\epsilon} = \lambda \mathbf{I} \otimes \mathbf{I} + 2\mu \mathbb{I}$, or equivalently, $f_{1,1}(\nabla \delta \mathbf{u}) = (\partial \boldsymbol{\sigma} / \partial \boldsymbol{\epsilon}) : \nabla \delta \mathbf{u} = \lambda \text{tr}[\boldsymbol{\epsilon}(\delta \mathbf{u})] + 2\mu \boldsymbol{\epsilon}(\delta \mathbf{u})$, which is equivalent to (2) applied to an increment $\delta \mathbf{u}$ within a nonlinear iterative solver. While these minor simplifications are possible for linear problems in the present work, our implementation solves the problem as though it were nonlinear, and we will continue using the corresponding terminology.

3 p -Multigrid Preconditioning of Linear Isotropic Elasticity

Using the formulation in Eq. (8), we can compute the action of global Jacobian matrix on with arbitrary user defined polynomial order p . An iterative linear equation solver is required with matrix-free operators, which necessitates preconditioning, especially at higher-order. With unstructured meshes, a natural hierarchy of grids does not exist, so h -multigrid can be difficult to implement. Algebraic MultiGrid (AMG) is suitable for low order meshes for which the Jacobian matrix can be assembled. However, assembly of this matrix is prohibitively expensive for higher-order element meshes [27]. Therefore, we use geometric multigrid with p coarsening while adopting AMG as the coarse grid solver. A Chebyshev polynomial smoother based upon the operator diagonal [28] is called in the multigrid cycle.

In p -multigrid, grid transfer operations increase or decrease the polynomial order of the element basis functions, and these operations can be implemented in a matrix-free fashion via $\mathbf{f}_{0,0}$ of (8) with suitable basis evaluators N . The coarse-to-fine (ctof) basis operation, N_{ctof} , interpolates the DoFs on the nodes of a coarse grid element to the nodes of a fine grid element ($N_{27 \times 8}$ for Q_1 prolongation to Q_2 , for example, where Q_1 denotes $p = 1$ and $Q_2, p = 2$). The corresponding coarse and fine grid element restriction operators, $\mathcal{E}_{e,c}$ and $\mathcal{E}_{e,f}$, are used in the grid transfer operators. The operator $\mathbf{P} = \sum_e \mathcal{E}_{e,f}^T N_{\text{ctof}} \mathcal{E}_{e,c}$ correctly computes the interior degrees of freedom but over-counts nodes on the element facets. We can count the multiplicity of each node on the fine grid by applying the transpose fine grid restriction to the unit vector, $\boldsymbol{\omega} = \sum_e \mathcal{E}_{e,f}^T \mathbf{1}$. Thus, the p -multigrid prolongation operator is given by,

$$\mathbf{P} = \sum_e \mathcal{E}_{e,f}^T \boldsymbol{\omega}^{-1} N_{\text{ctof}} \mathcal{E}_{e,c}, \quad (10)$$

and then the p -multigrid restriction operator is given by $\mathbf{R} = \mathbf{P}^T$.

4 Numerical Examples

Using the 3D linear isotropic elasticity model and matrix-free, higher-order FEM presented in Sections 2 and 3, we present numerical results based on (i) method of manufactured solutions (MMS) applied to a tube geometry, (ii) a tube bending simulation for compressible and nearly-incompressible elasticity on unstructured meshes, and (iii) a beam bending simulation with stress concentration on structured meshes. Polynomial orders $p = 1, 2, 3, 4$ are applied for a range of meshes in the compressible case, and polynomial orders $p = 2, 3, 4$ are applied for the same range of meshes in the nearly-incompressible case. For the compressible case, Poisson's ratio $\nu = 0.3$ and Young's modulus $E = 69 \times 10^9$ Pa, that correspond to aluminum. For the nearly-incompressible case, $\nu = 0.499999$ and $E = 0.1 \times 10^9$ Pa, that correspond to rubber-like materials in the small elastic strain regime.

The parallel platform used to run all FE simulations with higher-order polynomial element meshes is an Intel Xeon E5-2680 v3 @2.50 GHz (2 CPUs/node, 24 cores/node) and 113 GB RAM per compute node [29]. The PETSc toolkit v3.14 is called for mesh management, domain decomposition, parallel assembly operations, and communication over all MPI processes. The matrix-free Jacobian operator is implemented using libCEED v0.8 and PETSc for Eq. (8). Preconditioning with p -multigrid and AMG is enabled through PETSc's multigrid interface with grid-transfer and operator application at each level implemented in libCEED. Vectorized tensor-product operations for 8-element batches in the matrix-free operator evaluation on AVX-2 extensions of $\times 86$ instruction sets for each local processor are conducted in libCEED. For simulations conducted with

MMS and tube bending, an unstructured mesh consisting of 400 Hex8 elements (trilinear) is generated in the Trelis [30] software. The mesh is then refined 5 times to generate 3,200, 10,800, 25,600, 50,000, and 86,400 element meshes, respectively.

4.1 Achieving L^2 Strain Energy Error via h - and p -Refinement in Tube Meshes Subjected to MMS

Considering the tube geometry in Fig. 1 with method of manufactured solutions (MMS)-generated displacement boundary conditions, the first four tube unstructured mesh refinements with 3,200, 10,800, 25,600, and 50,000 elements are used in the FE simulations for the compressible and nearly-incompressible cases. For the MMS, we produce a contrived, but smooth, right hand side ρ based on $\rho = [u_1, u_2, u_3]^T$ with $u_1 = e^{2x} \sin(3y) \cos(4z)/10^8$, $u_2 = e^{3x} \sin(4y) \cos(2z)/10^8$, and $u_3 = e^{4x} \sin(2y) \cos(3z)/10^8$. Every problem is run once to determine the L^2 global strain energy Φ^h error from the MMS. Tables 1a and 1b summarize the size of each problem in terms of degrees of freedom (“#DoF”) based on h - and p -refinement and the corresponding L^2 strain energy error based on MMS for the compressible case. Tables 2a and 2b summarize the size of each problem in terms of DoF based on h - and p -refinement and the corresponding L^2 strain energy error based on MMS for the nearly-incompressible case. Considering the smoothness of the MMS, and relatively high number of DoF for even the coarsest mesh (3009 elements), the strain energy error is already quite small prior to refinement. Although, for p -refinement, we see for the compressible case some convergence, while for the nearly-incompressible case, the error is small and stays small (but not as small as the compressible case, 10^{-10} vs. 10^{-7}) indicating stability for smooth deformations via MMS.

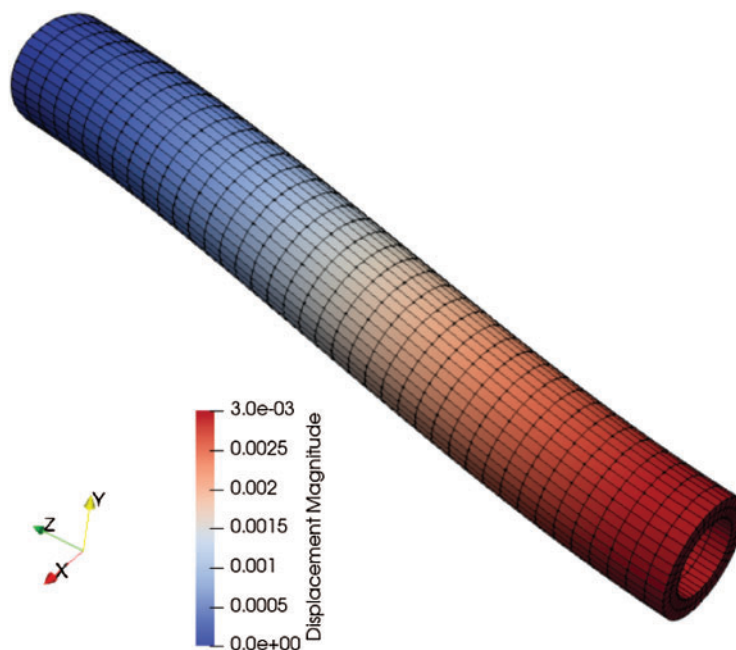


Figure 1: Deformed hexahedral mesh for the cylindrical tube bending problem. The length of the tube is 100 mm, with circular cross-section with inner diameter 15 mm and outer diameter 20 mm. The left wall is fixed in x , y , and z displacements, while the right wall is displaced in the negative y direction by 3 mm

Table 1: L^2 error (using MMS on tube geometry) for compressible elasticity ($\nu = 0.3$) with polynomial orders, $p = 1, 2, 3, 4$

deg	#Refine	#DoF	L^2 error
(a) L^2 error based on h -refinement			
1	1	3009	8.06243e - 06
1	2	9006	4.90064e - 06
1	3	20196	2.7557e - 06
1	4	43740	1.33188e - 06
2	1	41412	4.9902e - 09
2	2	103032	6.2258e - 10
2	3	212790	7.90538e - 10
2	4	439020	1.90668e - 10
3	1	159489	8.24201e - 11
3	2	382878	7.4593e - 11
3	3	776160	5.13348e - 11
3	4	1582416	1.91158e - 11
4	1	401520	4.96227e - 11
4	2	949344	9.59618e - 11
4	3	1908684	3.55511e - 11
4	4	3870504	6.72474e - 11
(b) L^2 error based on p -refinement			
1	1	3009	8.06243e - 06
2	1	41412	4.9902e - 09
3	1	159489	8.24201e - 11
4	1	401520	4.96227e - 11
1	2	9006	4.90064e - 06
2	2	103032	6.2258e - 10
3	2	382878	7.4593e - 11
4	2	949344	9.59618e - 11
1	3	20196	2.7557e - 06
2	3	212790	7.90538e - 10
3	3	776160	5.13348e - 11
4	3	1908684	3.55511e - 11
1	4	43740	1.33188e - 06
2	4	439020	1.90668e - 10
3	4	1582416	1.91158e - 11
4	4	3870504	6.72474e - 11

Table 2: L^2 error (using MMS on tube geometry) for nearly-incompressible elasticity ($\nu = 0.499999$) with polynomial orders, $p = 1, 2, 3, 4$

deg	#Refine	#DoF	L^2 Error
(a) L^2 error based on h -refinement			
1	1	3009	5.241318e – 05
1	2	9006	1.571074e – 04
1	3	20196	1.005251e – 04
1	4	43740	1.073758e – 04
2	1	41412	5.50025e – 07
2	2	103032	4.56178e – 07
2	3	212790	5.99263e – 07
2	4	439020	5.60685e – 07
3	1	159489	9.68488e – 07
3	2	382878	6.05481e – 07
3	3	776160	6.96476e – 07
3	4	1582416	5.67406e – 07
4	1	401520	1.05322e – 06
4	2	949344	9.16787e – 07
4	3	1908684	7.77346e – 07
4	4	3870504	6.70757e – 07
(b) L^2 error based on p -refinement			
1	1	3009	5.241318e – 05
2	1	41412	5.50025e – 07
3	1	159489	9.68488e – 07
4	1	401520	1.05322e – 06
1	2	9006	1.571074e – 04
2	2	103032	4.56178e – 07
3	2	382878	6.05481e – 07
4	2	949344	9.16787e – 07
1	3	20196	1.005251e – 04
2	3	212790	5.99263e – 07
3	3	776160	6.96476e – 07
4	3	1908684	7.77346e – 07
1	4	43740	1.073758e – 04
2	4	439020	5.60685e – 07
3	4	1582416	5.67406e – 07
4	4	3870504	6.70757e – 07

4.2 Achieving Reduced L^2 Strain Energy Error via h - and p -Refinement in Tube Meshes Subjected to Bending

Tube meshes subjected to bending are simulated to study cost and spatial convergence of the matrix-free, higher-order FEM for h -refinement and p -refinement based on strain energy Φ^h

for compressible and nearly-incompressible elasticity. The length of the tube is 100 mm, with circular cross-section of inner diameter 15 mm and outer diameter 20 mm. A coarse mesh representation of the tube is shown in Fig. 1. The left side of the tube is clamped by placing a zero displacement boundary condition on the surface, while a second displacement boundary condition is imposed on the right surface of the tube to bend it 3 mm in the negative y direction. Fig. 1 shows the deformed mesh after bending. All tube meshes are simulated for compressible and nearly-incompressible elasticity. Tables 3 and 4 summarize the problem sizes in terms of degrees of freedom (“#DoF”) for each refinement in conjunction with different polynomial orders p for compressible and nearly-incompressible elasticity, respectively. The size of problem (“#DoF”) increases when a finer mesh h or a higher-order polynomial p is used. As the FE mesh is refined in h , the norm of the strain energy computed from the FE solution converges to the exact norm of strain energy [31]. Therefore, we use the discrete elastic strain energy Φ^h to compute relative error in the FE simulations with respect to smaller size problems. Achieving a certain strain energy error in the FE solution is problem-dependent. For example, problems with smooth and non-smooth solutions behave differently with mesh refinement h when used with different polynomial orders p [18]. Therefore, with mesh refinement through decreasing h , we treat the strain energy from the finest mesh as the “exact” solution in the computation of relative L^2 error for each polynomial order p to determine the effect of h -refinement in achieving a certain error. However, with p -refinement, we treat the strain energy from the highest polynomial order ($p = 4$) to compute the relative L^2 strain energy error for each p -refinement. Tables 3a and 3b summarize the relative L^2 error based on mesh refinement h and using higher-order polynomials p for the compressible tube bending FE simulations, respectively. In addition, depending on the problem, each simulation is run with 1, 6, 12, 24, 48, 96, 192, 384, and 768 CPU cores for the compressible case. Each simulation is run 3 times to reduce noise in the performance timing for a total of 297 simulations for the compressible case.

Table 3: L^2 error based on strain energy Φ^h for compressible ($\nu = 0.3$) elastic tube bending

p	#Refine	#DoF	Strain Energy	L^2 Error
(a) L^2 error based on h -refinement				
1	1	14040	3.600109e + 00	2.2596e - 02
1	2	42480	3.554248e + 00	9.5695e - 03
1	3	94800	3.535690e + 00	4.2980e - 03
1	4	178200	3.526167e + 00	1.5933e - 03
1	5	299880	3.520558e + 00	0.0000e + 00
2	1	94800	3.483284e + 00	5.4041e - 03
2	2	299880	3.495535e + 00	1.9059e - 03
2	3	686880	3.499542e + 00	7.6197e - 04
2	4	1313400	3.501305e + 00	2.5850e - 04
2	5	2237040	3.502210e + 00	0.0000e + 00
3	1	299880	3.475215e + 00	7.3139e - 03
3	2	966600	3.491421e + 00	2.6847e - 03
3	3	2237040	3.496988e + 00	1.0946e - 03
3	4	4305600	3.499495e + 00	3.7828e - 04
3	5	7366680	3.500819e + 00	0.0000e + 00

(Continued)

Table 3 (continued)

p	#Refine	#DoF	Strain Energy	L^2 Error
(a) L^2 error based on h -refinement				
4	1	686880	3.473070e + 00	7.7888e - 03
4	2	2237040	3.490239e + 00	2.8839e - 03
4	3	5206080	3.496170e + 00	1.1896e - 03
4	4	10054800	3.498879e + 00	4.1542e - 04
4	5	17244000	3.500334e + 00	0.0000e + 00
#Refine	p	#DoF	Strain energy	L^2 Error
(b) L^2 error based on p -refinement				
1	1	14040	3.600109e + 00	3.6578e - 02
1	2	94800	3.483284e + 00	2.9409e - 03
1	3	299880	3.475215e + 00	6.1749e - 04
1	4	686880	3.473070e + 00	0.0000e + 00
2	1	42480	3.554248e + 00	1.8339e - 02
2	2	299880	3.495535e + 00	1.5174e - 03
2	3	966600	3.491421e + 00	3.3857e - 04
2	4	2237040	3.490239e + 00	0.0000e + 00
3	1	94800	3.535690e + 00	1.1304e - 02
3	2	686880	3.499542e + 00	9.6448e - 04
3	3	2237040	3.496988e + 00	2.3394e - 04
3	4	5206080	3.496170e + 00	0.0000e + 00
4	1	178200	3.526167e + 00	7.7990e - 03
4	2	1313400	3.501305e + 00	6.9319e - 04
4	3	4305600	3.499495e + 00	1.7597e - 04
4	4	10054800	3.498879e + 00	0.0000e + 00
5	1	299880	3.520558e + 00	5.7779e - 03
5	2	2237040	3.502210e + 00	5.3612e - 04
5	3	7366680	3.500819e + 00	1.3882e - 04
5	4	17244000	3.500334e + 00	0.0000e + 00

To address the first question raised in the Introduction with regard to minimal computational cost for a certain strain energy error in the FE solution for compressible elasticity, we compute the minimum amount of CPU work to reach a certain error based on mesh refinements and higher-order polynomials. To do this, for each polynomial order p we accumulate the time spent to reach the error. Then we determine the minimum time in the list and multiply it by the number of processors np that are executed for the minimum time. We generate *Pareto optimal* diagrams to present the results of the study for compressible and nearly-incompressible elasticity separately, where a point on the diagram is called *Pareto optimal* if strain energy error cannot be decreased (moving down the ordinate (y axis)) without increasing time (or cost) by moving to the right on the abscissa. The set of Pareto optimal configurations is known as the *Pareto front*.

Table 4: L^2 error based on strain energy Φ^h for nearly-incompressible ($\nu = 0.499999$) elastic tube bending

p	#Refine	#DoF	Strain Energy	L^2 Error
(a) L^2 error based on h -refinement				
2	1	94800	1.368802e + 01	6.6788e - 01
2	2	299880	2.057905e + 01	5.0069e - 01
2	3	686880	2.746070e + 01	3.3371e - 01
2	4	1313400	3.433859e + 01	1.6683e - 01
2	5	2237040	4.121459e + 01	0.0000e + 00
3	1	299880	2.542062e + 01	6.6788e - 01
3	2	966600	3.821824e + 01	5.0069e - 01
3	3	2237040	5.099845e + 01	3.3371e - 01
3	4	4305600	6.377166e + 01	1.6683e - 01
3	5	7366680	7.654139e + 01	0.0000e + 00
4	1	686880	4.106407e + 01	6.6788e - 01
4	2	2237040	6.173716e + 01	5.0069e - 01
4	3	5206080	8.238211e + 01	3.3371e - 01
4	4	10054800	1.030158e + 02	1.6683e - 01
4	5	17244000	1.236438e + 02	0.0000e + 00
#Refine	p	#DoF	Strain Energy	L^2 Error
(b) L^2 error based on p -refinement				
1	2	94800	1.368802e + 01	6.6667e - 01
1	3	299880	2.542062e + 01	3.8095e - 01
1	4	686880	4.106407e + 01	0.0000e + 00
2	2	299880	2.057905e + 01	6.6667e - 01
2	3	966600	3.821824e + 01	3.8095e - 01
2	4	2237040	6.173716e + 01	0.0000e + 00
3	2	686880	2.746070e + 01	6.6667e - 01
3	3	2237040	5.099845e + 01	3.8095e - 01
3	4	5206080	8.238211e + 01	0.0000e + 00
4	2	1313400	3.433859e + 01	6.6667e - 01
4	3	4305600	6.377166e + 01	3.8095e - 01
4	4	10054800	1.030158e + 02	0.0000e + 00
5	2	2237040	4.121459e + 01	6.6667e - 01
5	3	7366680	7.654139e + 01	3.8095e - 01
5	4	17244000	1.236438e + 02	0.0000e + 00

Figs. 2a and 2b present the L^2 strain energy error based on mesh refinement *vs.* cost, and the L^2 error based on higher-order polynomials *vs.* cost, for compressible elasticity. Figs. 2a and 2b show that the minimal cost to achieve a certain error occurs in the lower left region of the diagram, indicating that higher order p is the most cost-effective for the compressible case. Each horizontal series of the same color represents a strong scaling study at fixed mesh refinement h and polynomial order p , but with changing number of processors np , with perfect strong scaling occurring when all color dots are collocated. The more expensive simulations tend to exhibit better strong scaling because they have more work over which to amortize the inherent communication costs, while smaller models (fewer DoFs) are more cost-efficient to run on a single core. The low order $p = 1$ cases are increasingly far from the Pareto front. In addition, solely calling larger number of CPU cores is less cost-efficient, although are faster with respect to solve time up to the point where the parallel overhead communication begin to dominate.

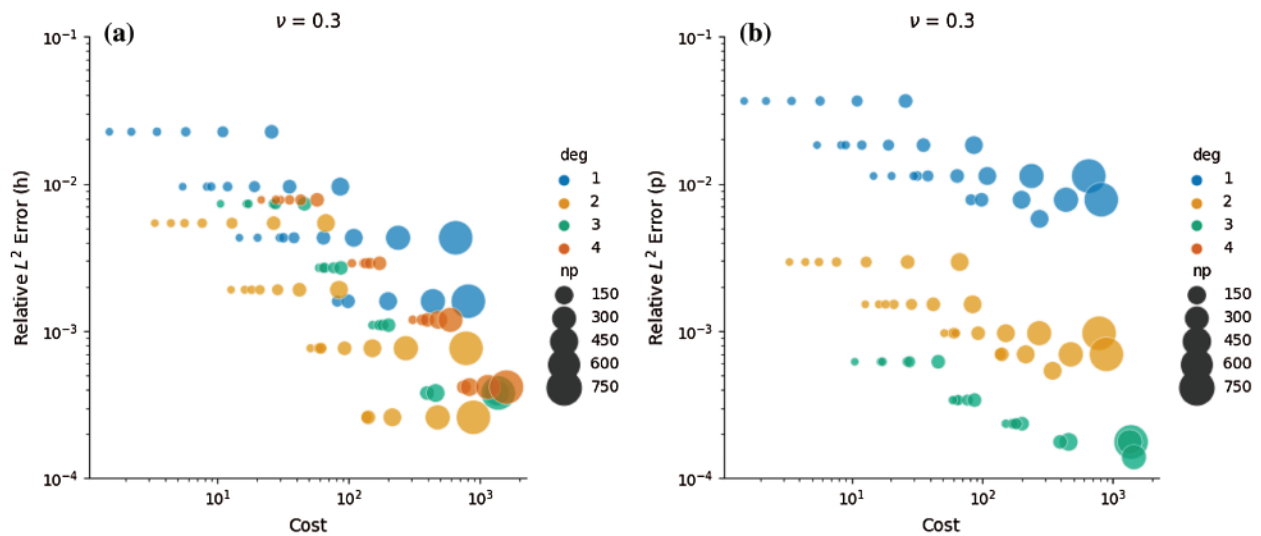


Figure 2: Error *vs.* cost for compressible elastic tube bending. The Pareto optimal configurations occur in the lower left region of the plot indicating a higher-order polynomial p is more cost-efficient for a certain error. (a) Error based on h -refinement *vs.* cost. (b) Error based on p -refinement *vs.* cost

To answer the second question raised in the Introduction regarding the least amount of time to achieve a certain strain energy error given enough computational resources, we provide a second set of Pareto optimal diagrams in Figs. 3a and 3b for the compressible elasticity tube bending problem. The Pareto optimal configurations are toward the lower left region of the plot, whereby upon p -refinement, $p = 3$ exhibits faster solve times for certain error, while upon h -refinement, $p = 2$ delivers the fastest solve times for certain error, a result that was unexpected and requires further investigation. Each horizontal series of the same color represents strong scaling of a given mesh refinement h and polynomial order p . Larger number of CPU cores when enough work is amortized tend to reach a certain error in the solution faster with higher order polynomials. Similar to Pareto optimal for the cost, strong scaling occurs when the same color larger dots overlay smaller dots.

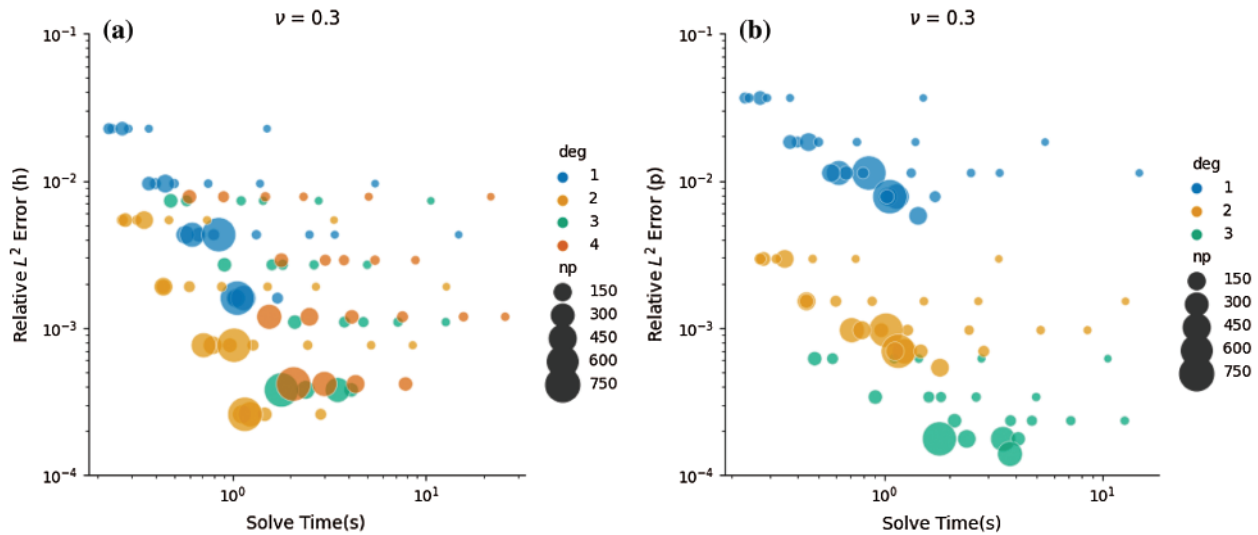


Figure 3: Error vs. time for compressible elastic tube bending. The Pareto optimal configurations occur toward the lower left region of the plot. (a) Error based on h -refinement vs. time. (b) Error based on p -refinement vs. time

We notice that depending on the error, increasing the polynomial order p while fixing mesh size (h constant) may provide the fastest time to solution. For example, for strain energy error between 10^{-3} and 10^{-4} , polynomial order $p = 3$ may be faster than lower order polynomials, but for error greater than 10^{-3} , polynomial order $p = 2$ may be faster. This could be due to the number of DoFs for different polynomial orders when run with the same size mesh. In addition, we expected that higher order polynomials $p = 3, 4$ would be more cost-efficient than lower order polynomials. However, we observed that polynomial order $p = 2$ is more cost-efficient than higher order polynomials $p = 3, 4$ for this compressible tube bending problem.

For nearly-incompressible elasticity, we use the same meshes h and polynomial order p . The problem sizes in terms of DoFs and L^2 strain energy error based on mesh refinement and polynomial orders are represented in Tables 4a and 4b. However, we instead use 24, 48, 96, 192, 384, 768, and 1536 CPU cores since nearly-incompressible elasticity is more computationally intensive for matrix-free solution. This resulted in 228 FE simulations. Figs. 4a and 4b represent Pareto optimal diagrams for the L^2 error based on mesh refinement vs. cost, and the L^2 error based on higher-order polynomials vs. cost, respectively. Pareto optimal consists of polynomial orders $p = 2, 3$ when L^2 error is computed based on h -refinement, while polynomial order $p = 3$ is more cost-effective in achieving more accuracy in the solution. Similarly, the Pareto optimal diagrams are provided for L^2 error based on mesh refinement h vs. solve time, and L^2 error based on polynomial order p vs. solve time in Figs. 5a and 5b, respectively. The Pareto front consists of $p = 2, 3, 4$ in that order. In addition, larger number of CPU cores provide faster solutions for a certain strain energy error.

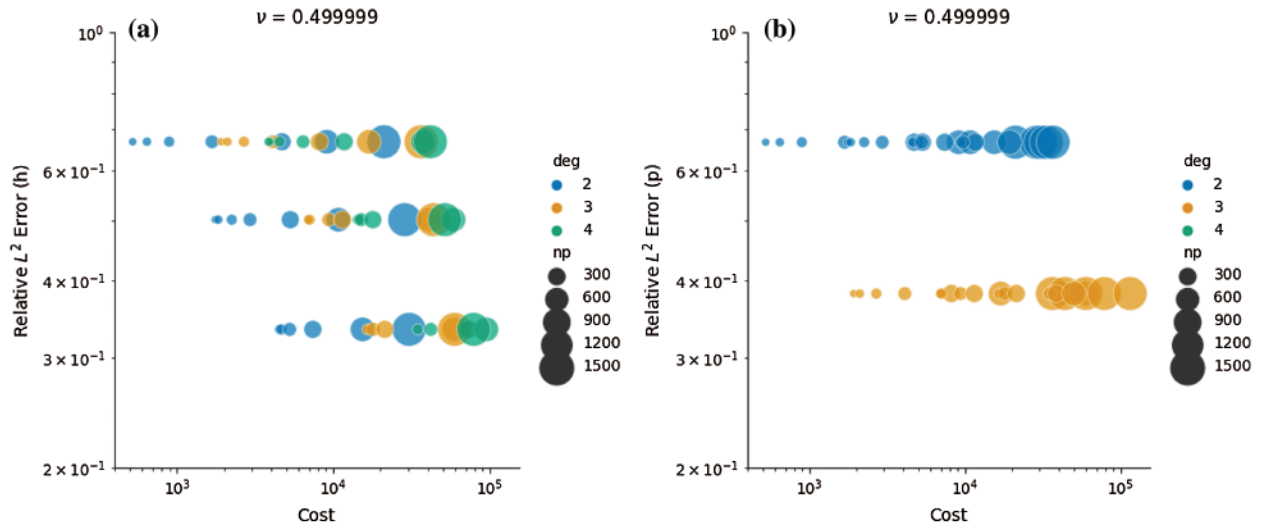


Figure 4: Error vs. cost for nearly-incompressible elastic tube bending. The Pareto optimal configurations occur toward the left region of the plot. Perfect strong scaling occurs when same color dots are collocated. (a) Error based on h -refinement vs. cost. (b) Error based on p -refinement vs. cost

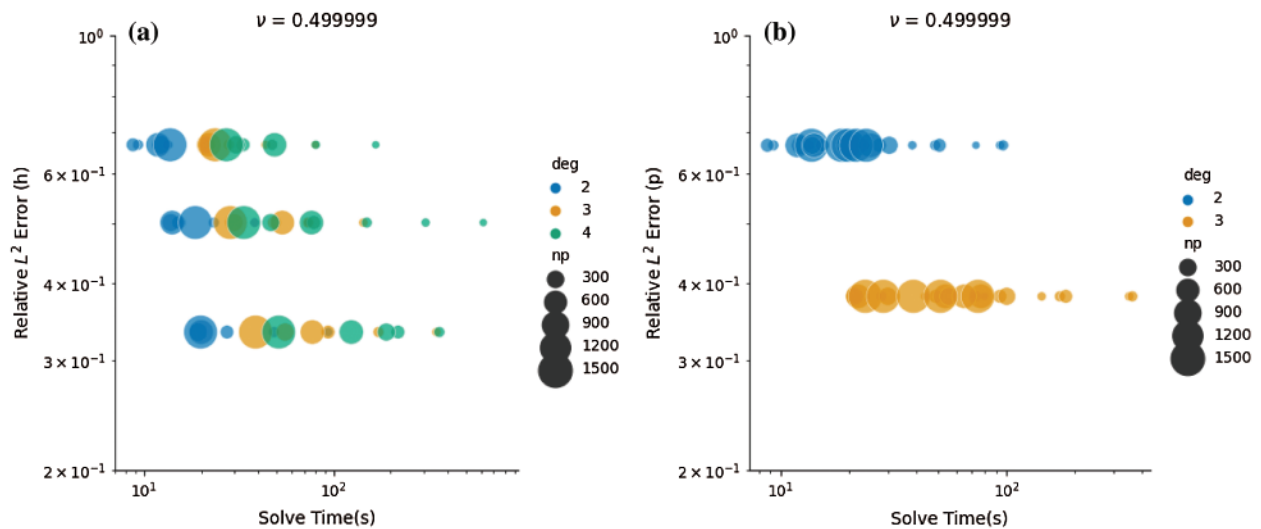


Figure 5: Error vs. time for nearly-incompressible elastic tube bending. The Pareto optimal configurations occur toward the left region of the plot. Perfect strong scaling occurs when same color dots are collocated. (a) Error based on h -refinement vs. time. (b) Error based on p -refinement vs. time

Next, we examine the relative parallel scalability of the matrix-free FE implementation in terms of throughput vs. solve time for $p = 1, 2, 3, 4$ in the compressible case, and $p = 2, 3, 4$ in the nearly-incompressible case. Using the tube geometry, new meshes are generated in Trellis. A mesh with 1,638,400 elements is used with polynomial order $p = 1$. Meshes with 204,800,

60,500, and 25,600 elements are used with polynomial order $p = 2, 3, 4$, respectively. These mesh sizes in conjunction with polynomial orders $p = 1, 2, 3, 4$ produce approximately 5.2M DoFs. This problem size is conveniently analyzed in the Pareto diagrams, which roughly occupies one compute node (24 CPU cores) on the parallel platform. For the compressible case, depending on the number of elements, 12, 24, 48, 96, 192, 384, and 768 CPU cores are executed where every problem is run three times for a total of 75 FE simulations. For the nearly-incompressible case, 192, 384, 768, and 1536 CPU cores are called. This resulted in 42 FE simulations for which each problem is run three times. Throughput is computed as millions of DoFs per second per CPU core for each FE simulation. The result of throughput *vs.* solve time for the compressible and nearly-incompressible cases are presented in Figs. 6a and 6b, respectively. We notice for the compressible case, the implementation scales up to 200–250 CPU cores (Fig. 7a) with polynomial orders 3 and 4. The sub-optimal performance in the code pertains to lack of work amortized to each CPU core for the problem size chosen. However, for the nearly-incompressible case, which is more computationally-intensive compared to the compressible case, the implementation scales up to 800–1000 CPU cores (Fig. 7b) for polynomial orders 2–4 for the problems sizes selected.

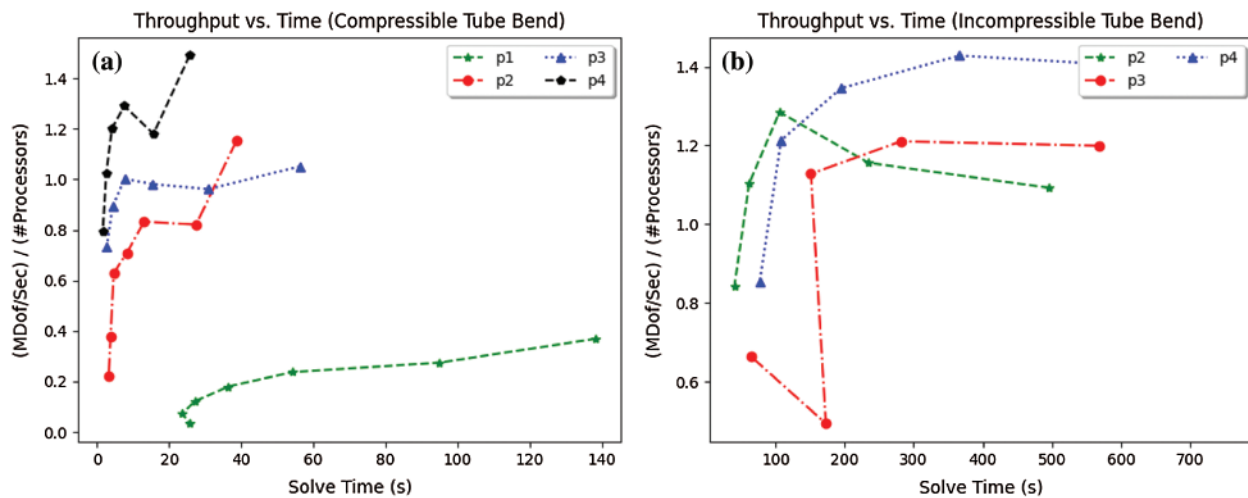


Figure 6: Throughput *vs.* solve time for compressible and nearly-incompressible elasticity problems. For the problem size with 5.2 M DoFs across polynomial orders $p = 1, 2, 3, 4$. (a) Throughput *vs.* solve time. (b) Throughput *vs.* solve time

4.3 Beam Bending under Body Force

In this section, we consider a solid square-cross-section aluminum beam subject to a body force in the negative y direction simulated with $p = 1, 2, 3, 4$. Fig. 8 shows a coarse structured mesh of the beam. The length of the beam is 5 m, and each side of the square beam cross-section is 0.25 m. The beam is clamped on both ends. A body force of 200 N/m^3 is applied in the negative y direction on the beam, which causes the beam to bend 9.1×10^{-4} mm in the negative y direction at the middle of the beam. Fig. 8 shows the deformed mesh of the beam. Similar to the tube bending simulation, the clamped BCs will generate stress concentration (singularity) in the clamped regions. When a stress concentration exists, it can be shown that the order of convergence of the FE solution becomes independent of polynomial order p [32]. That is, the order of convergence of the FE solution cannot be improved by using higher-order finite

elements. We investigate this phenomenon by conducting h - and p -refinement with our matrix-free FE implementation in libCEED/PETSc.

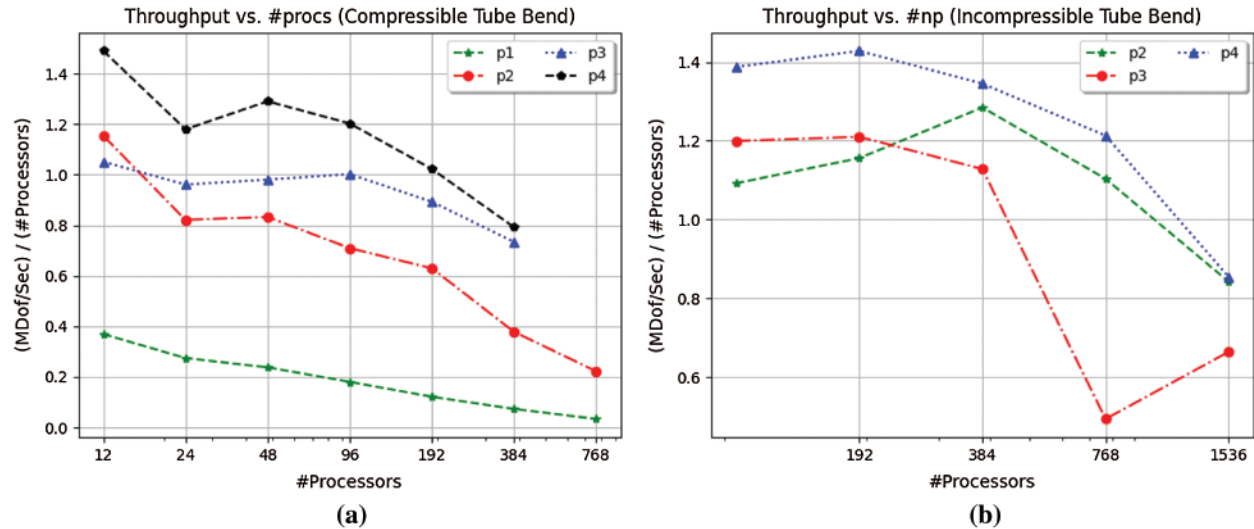


Figure 7: Throughput vs. number of processors for compressible and nearly-incompressible elasticity problems. For the problem size with 5.2 M DoFs across polynomial orders $p = 1, 2, 3, 4$, the implementation remains scalable up to 200–250 CPU cores in the compressible case with polynomial order 3 and 4, while it is scalable up to 800–1000 CPU cores in the more computationally-intensive nearly-incompressible case. (a) Throughput vs. solve time. (b) Throughput vs. solve time

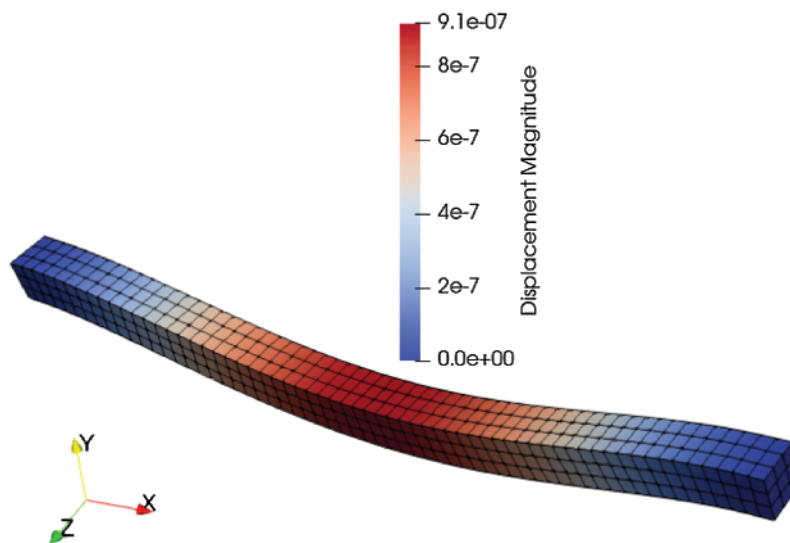


Figure 8: Deformed hexahedral beam structured mesh 5 m in length and square cross-section with side 0.25 m. Beam bent by 200 N/m^3 body force in the negative y direction, while the left and right ends are clamped. The body force causes a $9.1 \times 10^{-4} \text{ mm}$ displacement in the middle of the beam

Five different refinements of the Hex8 beam mesh are generated with the Trellis software for 558, 4,464, 15,066, 35,712, and 69,750 elements. The finest mesh with 69,750 elements acts as the “exact” solution with respect to strain energy calculation. A total of 60 simulations are conducted where each simulation is repeated 3 times. An L^2 strain energy error based on h -refinement per polynomial order is computed for $p = 1, 2, 3, 4$. In addition, an L^2 strain energy error based on higher-order polynomial p for each mesh refinement h is computed. Tables 5a and 5b summarize the refinement (“#Refine”), polynomial degree p (“deg”), number of degrees of freedom (#DoF), strain energy, and the L^2 error. The element size for each mesh is computed to be $h = 0.1428, 0.0714, 0.0476, 0.0357$ m for smallest to largest number of elements mesh, respectively. Fig. 9 represents the h -refinement order of convergence for polynomial orders 1 through 4. The calculated slopes (order of convergence) of the lines that are best fit in the least square sense for polynomial orders $p = 1, 2, 3, 4$ with h -refinement are 2.43, 2.48, 2.38, and 2.31, respectively. Therefore, we conclude that for this problem the order of convergence of the FE solution on structured meshes with stress concentration (singularity) cannot be improved by using higher-order polynomials. Thus, the lack of optimal spatial convergence with regard to higher-order FE when there is a stress concentration is confirmed, as opposed to when there is no stress concentration (or singularity) [22].

Table 5: L^2 error based on strain energy Φ^h for compressible elasticity ($\nu = 0.3$) beam bending under body force for polynomial orders, $p = 1,2,3,4$

p	#Refine	#DoF	Strain Energy	L^2 Error
(a): L^2 error based on h -refinement				
1	0	2928	1.585403e – 05	2.9957e – 02
1	1	18081	1.621649e – 05	7.7793e – 03
1	2	55500	1.629746e – 05	2.8253e – 03
1	3	125229	1.632844e – 05	9.2984e – 04
1	4	237312	1.634363e – 05	0.0000e + 00
2	0	18081	1.633363e – 05	2.3822e – 03
2	1	125229	1.636326e – 05	5.7237e – 04
2	2	401793	1.636923e – 05	2.0805e – 04
2	3	928125	1.637149e – 05	6.9598e – 05
2	4	1784577	1.637263e – 05	0.0000e + 00
3	0	55500	1.636293e – 05	6.9572e – 04
3	1	401793	1.637133e – 05	1.8308e – 04
3	2	1310064	1.637321e – 05	6.8135e – 05
3	3	3051501	1.637394e – 05	2.3227e – 05
3	4	5897292	1.637432e – 05	0.0000e + 00
4	0	125229	1.636983e – 05	3.0598e – 04
4	1	928125	1.637342e – 05	8.7162e – 05
4	2	3051501	1.637431e – 05	3.2450e – 05
4	3	7138173	1.637466e – 05	1.1307e – 05
4	4	13830957	1.637484e – 05	0.0000e + 00

(Continued)

Table 5 (continued)

#Refine	p	#DoF	Strain Energy	L^2 Error
(b) L^2 error based on p -refinement				
0	1	2928	1.585403e - 05	3.1509e-02
0	2	18081	1.633363e - 05	2.2117e - 03
0	3	55500	1.636293e - 05	4.2167e - 04
0	4	125229	1.636983e - 05	0.0000e + 00
1	1	18081	1.621649e - 05	9.5842e - 03
1	2	125229	1.636326e - 05	6.2028e - 04
1	3	401793	1.637133e - 05	1.2775e - 04
1	4	928125	1.637342e - 05	0.0000e + 00
2	1	55500	1.629746e - 05	4.6936e - 03
2	2	401793	1.636923e - 05	3.1068e - 04
2	3	1310064	1.637321e - 05	6.7510e - 05
2	4	3051501	1.637431e - 05	0.0000e + 00
3	1	125229	1.632844e - 05	2.8228e - 03
3	2	928125	1.637149e - 05	1.9338e - 04
3	3	3051501	1.637394e - 05	4.3745e - 05
3	4	7138173	1.637466e - 05	0.0000e + 00
4	1	237312	1.634363e - 05	1.9060e - 03
4	2	1784577	1.637263e - 05	1.3509e - 04
4	3	5897292	1.637432e - 05	3.1825e - 05
4	4	13830957	1.637484e - 05	0.0000e + 00

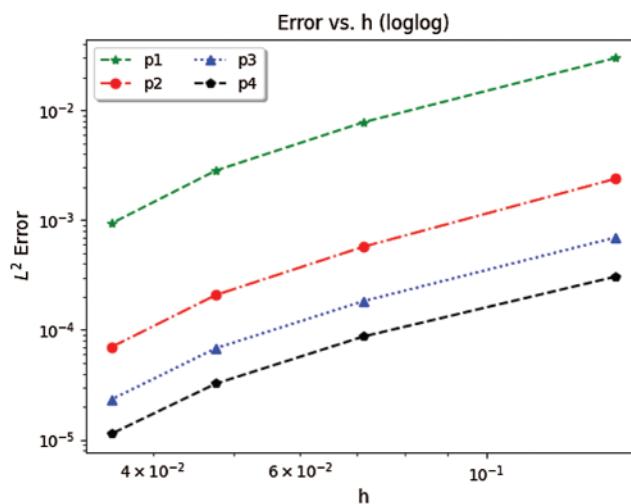


Figure 9: h -refinement order of convergence plot for clamped beam bending under body force with polynomials $p = 1, 2, 3, 4$

5 Conclusions

The paper investigated the use of displacement-based higher-order finite elements for compressible and nearly-incompressible linear isotropic elasticity using a matrix-free implementation with p -multigrid preconditioning. It was observed that varying the polynomial order p and mesh size h affects the cost and time it takes to achieve a certain strain energy error for Poisson's ratios $\nu = 0.3, 0.499999$ in a tube bending problem. Increasing the polynomial order p while fixing the mesh size (h constant) may provide the fastest time to converge to a certain strain energy error between 10^{-3} and 10^{-4} . However, for strain energy error greater than 10^{-3} , lower-order polynomials may achieve the solution faster. When there is a stress concentration present, higher-order polynomials do not improve the order of convergence in the finite element solution. In addition, polynomial order $p = 2$ proved to be more efficient than $p = 3, 4$ for the compressible elastic tube bending problem, which also has a stress concentration (singularity in the stress solution). Each mesh refinement was run with polynomial orders $p = 1, 2, 3, 4$ for compressible elasticity, and polynomial orders $p = 2, 3, 4$ for nearly-incompressible elasticity. Using the same mesh refinements h with higher-order polynomials ($p = 3, 4$) produce a much larger problem size (more DoFs) than when the same meshes are used with polynomial order 2. Therefore, polynomial orders $p = 3, 4$ appear to be more costly than polynomial order $p = 2$. An exhaustive search using different polynomial orders p and mesh sizes h may determine if polynomial order $p = 2$ is truly more cost efficient than polynomial orders $p = 3, 4$, which will be considered as future work. In addition, for nearly-incompressible elasticity ($\nu = 0.499999$), the tube bending problem could not achieve an error better than 10^{-1} based on discrete global strain energy Φ^h computation because of the stress concentration; whereas a smooth method of manufactured solution (MMS) tube problem achieved strain energy error on the order of 10^{-7} , illustrating the difficulty in convergence for h - and p -refinement for near-incompressibility when a stress concentration is present. Therefore, implementing a mixed formulation for handling $\nu \rightarrow 0.5$ appears to be expedient with respect to spatial convergence than solely a displacement-based, higher-order finite element method. Scalability of the matrix-free FEM for compressible elasticity is feasible up to a few hundred CPU cores, while scalability improves for the more computationally-expensive nearly-incompressible case to one thousand CPU cores for the problem sizes chosen in this study. Higher-order finite elements generally still outperform lower-order finite elements within the matrix-free implementation presented in the paper, for which the cost per degree of freedom decreases with increasing p , due to more structured computation and more efficient quadrature.

Notation: Boldface denotes vectors and tensors in symbolic form. Unless otherwise indicated, all vector and tensor products in symbolic form are assumed to be inner products, such as $\mathbf{v}\mathbf{v} = v_i v_i$, $(\mathbf{ab})_{ik} = a_{ij} b_{jk}$ and $(\mathbf{a}) : (\mathbf{b}) = a_{ij} b_{ij}$, where repeated indices denote a sum over those indices. Cartesian coordinates are assumed. The symbol $\text{tr}(\bullet)$ is the trace operator, such that $\text{tr}(\boldsymbol{\sigma}) = \sigma_{ii}$. The symbol \mathbf{I} is the unit tensor, i.e., $\mathbf{I}_{ij} = \delta_{ij}$, where δ_{ij} is the Kronecker delta operator, and \mathbb{I} is the fourth order identity tensor. p refers to polynomial basis order, h to characteristic element size, d to spatial dimension, and np to number of processors.

Funding Statement: The research relied on computational resources [29] provided by the University of Colorado Boulder Research Computing Group, which is supported by the National

Science Foundation (Awards ACI-1532235 and ACI-1532236), University of Colorado Boulder, and Colorado State University.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Zienkiewicz, O. C., Taylor, R. L., Zhu, J. Z. (2005). *The finite element method: Its basis and fundamentals*. Amsterdam, Netherlands: Elsevier.
2. Hughes, T. J. R. (2000). *The finite element method: Linear static and dynamic finite element analysis*. Mineola, NY: Dover Publications, Inc.
3. Fraeijns de Veubeke, B. (1965). Displacement and equilibrium models in the finite element method. *Stress analysis*, pp. 145–197. Hoboken, New Jersey: John Wiley & Sons.
4. Chiumenti, M., Valverde, Q., de Saracibar, C. A., Cervera, M. (2002). A stabilized formulation for incompressible elasticity using linear displacement and pressure interpolations. *Computer Methods in Applied Mechanics and Engineering*, 191(46), 5253–5264. DOI 10.1016/S0045-7825(02)00443-7.
5. Nakshatrala, K., Masud, A., Hjelmstad, K. (2008). On finite element formulations for nearly incompressible linear elasticity. *Computational Mechanics*, 41(4), 547–561. DOI 10.1007/s00466-007-0212-8.
6. Tchonkova, M., Peters, J., Sture, S. (2011). Three-dimensional modeling of problems in poro-elasticity via a mixed least squares method using linear tetrahedral elements. *International Journal for Numerical and Analytical Methods in Geomechanics*, 35(15), 1656–1681. DOI 10.1002/nag.971.
7. Pian, T. H., Sumihara, K. (1984). Rational approach for assumed stress finite elements. *International Journal for Numerical Methods in Engineering*, 20(9), 1685–1695. DOI 10.1002/(ISSN)1097-0207.
8. Hughes, T. J., Franca, L. P., Balestra, M. (1986). A new finite element formulation for computational fluid dynamics: V. circumventing the Babuška–Brezzi condition: A stable petrov–galerkin formulation of the stokes problem accommodating equal-order interpolations. *Computer Methods in Applied Mechanics and Engineering*, 59(1), 85–99. DOI 10.1016/0045-7825(86)90025-3.
9. Sussman, T., Bathe, K. J. (1985). Studies of finite element procedures on mesh selection. *Computers & Structures*, 21(1–2), 257–264. DOI 10.1016/0045-7949(85)90248-2.
10. Brown, J. (2010). Efficient nonlinear solvers for nodal high-order finite elements in 3D. *Journal of Scientific Computing*, 45(1–3), 48–63. DOI 10.1007/s10915-010-9396-8.
11. Deville, M. O., Fischer, P. F., Mund, E. H. (2002). *High-order methods for incompressible fluid flow*, pp. 528. Cambridge: Cambridge University Press.
12. Knoll, D. A., Keyes, D. E. (2004). Jacobian-free Newton–Krylov methods: A survey of approaches and applications. *Journal of Computational Physics*, 193(2), 357–397. DOI 10.1016/j.jcp.2003.08.010.
13. Saad, Y. (2003). *Iterative methods for sparse linear systems*. 2nd ed. Philadelphia: Society for Industrial and Applied Mathematics.
14. Zhu, Y., Sifakis, E., Teran, J., Brandt, A. (2010). An efficient multigrid method for the simulation of high-resolution elastic solids. *ACM Transactions on Graphics*, 29(2), 16–18. DOI 10.1145/1731047.1731054.
15. May, D., Brown, J., Pourhiet, L. L. (2015). A scalable, matrix-free multigrid preconditioner for finite element discretizations of heterogeneous stokes flow. *Computer Methods in Applied Mechanics and Engineering*, 290, 496–523. DOI 10.1016/j.cma.2015.03.014.
16. Kronbichler, M., Ljungkvist, K. (2019). Multigrid for matrix-free high-order finite element computations on graphics processors. *ACM Transactions on Parallel Computing*, 6(1), 1–32. DOI 10.1145/3322813.
17. Rønquist, E. M., Patera, A. T. (1987). Spectral element multigrid. I. Formulation and numerical results. *Journal of Scientific Computing*, 2(4), 389–406. DOI 10.1007/BF01061297.
18. Vos, P. E., Sherwin, S. J., Kirby, R. M. (2010). From h to p efficiently: Implementing finite and spectral/hp element methods to achieve optimal performance for low-and high-order discretisations. *Journal of Computational Physics*, 229(13), 5161–5181. DOI 10.1016/j.jcp.2010.03.031.

19. Cantwell, C. D., Sherwin, S. J., Kirby, R. M., Kelly, P. H. (2011). From h to p efficiently: Selecting the optimal spectral/hp discretisation in three dimensions. *Mathematical Modelling of Natural Phenomena*, 6(3), 84–96. DOI 10.1051/mmnp/20116304.
20. Remacle, J. F., Flaherty, J. E., Shephard, M. S. (2003). An adaptive discontinuous galerkin technique with an orthogonal basis applied to compressible flow problems. *SIAM Review*, 45(1), 53–72. DOI 10.1137/S00361445023830.
21. Herrmann, L. R. (1965). Elasticity equations for incompressible and nearly incompressible materials by a variational theorem. *AIAA Journal*, 3(10), 1896–1900. DOI 10.2514/3.3277.
22. Mehraban, A., Thompson, J., Brown, J., Regueiro, R., Barra, V. et al. (2020). Simulating compressible and nearly-incompressible linear elasticity using an efficient parallel scalable matrix-free high-order finite element method. *14th WCCM-ECCOMAS Congress*. DOI 10.23967/wccm-eccomas.2020.302.
23. Mehraban, A. (2021). *Efficient residual and matrix-free jacobian evaluation for three-dimensional hexahedral finite elements with nearly-incompressible neo-hookean hyperelasticity as applied to soft materials (Ph.D. thesis)*. University of Colorado Boulder.
24. Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P. et al. (2020). PETSc users manual. <https://www.mcs.anl.gov/petsc>.
25. Abdelfattah, A., Barra, V., Beams, N., Brown, J., Camier, J. S. et al. (2020). libCEED User Manual. <https://doi.org/10.5281/zenodo.4302737>.
26. Knepley, M. G., Brown, J., Rupp, K., Smith, B. F. (2013). Achieving high performance with unified residual evaluation. <https://arxiv.org/abs/1309.1204v2>.
27. Heys, J., Manteuffel, T., McCormick, S. F., Olson, L. (2005). Algebraic multigrid for higher-order finite elements. *Journal of Computational Physics*, 204(2), 520–532. DOI 10.1016/j.jcp.2004.10.021.
28. Adams, M., Brezina, M., Hu, J., Tuminaro, R. (2003). Parallel multigrid smoothing: Polynomial vs. Gauss–Seidel. *Journal of Computational Physics*, 188(2), 593–610. DOI 10.1016/S0021-9991(03)00194-3.
29. Anderson, J., Burns, P. J., Milroy, D., Ruprecht, P., Hauser, T. et al. (2017). Deploying RMACC summit: An HPC resource for the rocky mountain region. *Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact*. New York, NY, USA: Association for Computing Machinery.
30. Coreform Cubit. <https://coreform.com/products/coreform-cubit/>.
31. Mahomed, N., Kekana, M. (1998). An error estimator for adaptive mesh refinement analysis based on strain energy equalisation. *Computational Mechanics*, 22(4), 355–366. DOI 10.1007/s004660050367.
32. Pin, T., Pian, T. H. (1973). On the convergence of the finite element method for problems with singularity. *International Journal of Solids and Structures*, 9(3), 313–321. DOI 10.1016/0020-7683(73)90082-6.