

**EFFICIENT RESIDUAL AND MATRIX-FREE JACOBIAN EVALUATION FOR  
THREE-DIMENSIONAL TRI-QUADRATIC HEXAHEDRAL FINITE ELEMENTS WITH  
NEARLY-INCOMPRESSIBLE NEO-HOOKEAN HYPERELASTICITY APPLIED TO  
SOFT MATERIALS ON UNSTRUCTURED MESHES IN PARALLEL,  
WITH PETSC AND LIBCEED**

**Arash Mehraban,\***

**Jed Brown, Valeria Barra,  
Henry Tufo**

Department of Computer Science  
University of Colorado Boulder  
Boulder, CO 80309  
arash.mehraban@colorado.edu,  
jed.brown@colorado.edu,  
valeria.barra@colorado.edu,  
henry.tufo@colorado.edu

**Jeremy Thompson**

Department of Applied Mathematics  
University of Colorado Boulder  
Boulder, CO 80309  
jeremy.thompson@colorado.edu

**Richard Regueiro**

Department Civil, Environmental,  
and Architectural Engineering  
University of Colorado Boulder  
Boulder, CO 80309  
richard.regueiro@colorado.edu

**ABSTRACT**

*Soft materials such as rubber, elastomers, and soft biological tissues mechanically deform at large strain isochorically for all time, or during their initial transient (when a pore fluid, typically incompressible such as water, does not have time to flow out of the deforming polymer or soft tissue porous skeleton). Simulating these large isochoric deformations computationally, such as with the Finite Element Method (FEM), requires higher order (typically quadratic) interpolation functions and/or enhancements through hybrid/mixed methods to maintain stability. Lower order (linear) finite elements with hybrid/mixed formulation may not perform stably for all mechanical loading scenarios involving large isochoric deformations, whereas quadratic finite elements with or without hybrid/mixed formulation typically perform stably, especially when large bending or folding deformations are being simulated. For topology-optimization design of soft robotics, for instance, the FEM solid mechanics solver must run efficiently and stably. Stability is ensured by the higher order finite element formulation (with possible enhancement), but effi-*

*ciency for higher order FEM remains a challenge. Thus, this paper addresses efficiency from the perspective of computer science algorithms and programming. The proposed efficient algorithm utilizes the Portable, Extensible Toolkit for Scientific Computation (PETSc), along with the libCEED library for efficient compiler optimized tensor-product-basis computation to demonstrate an efficient nonlinear solution algorithm. For preconditioning, a scalable  $p$ -multigrid method is presented whereby a hierarchy of levels is constructed. In contrast to classical geometric multigrid, also known as  $h$ -multigrid, each level in  $p$ -multigrid is related to a different approximation polynomial order,  $p$ , instead of the element size,  $h$ . A Chebyshev polynomial smoother is used on each multigrid level. Algebraic MultiGrid (AMG) is then applied to the assembled  $Q_1$  (linear) coarse mesh on the nodes of the quadratic  $Q_2$  (quadratic) mesh. This allows low storage that can be efficiently used to accelerate the convergence to solution. For a Neo-Hookean hyperelastic problem, we examine a residual and matrix-free Jacobian formulation of a tri-quadratic hexahedral finite element with enhancement. Efficiency estimates on AVX-2 architecture based on CPU time are provided as a comparison*

\*Address all correspondence to this author.

to similar simulation (and mesh) of isochoric large deformation hyperelasticity as applied to soft materials conducted with the commercially-available FEM software program ABAQUS. The particular problem in consideration is the simulation of an assistive device in the form of finger-bending in 3D.

## NOMENCLATURE

Boldface denotes vectors and tensors in symbolic notation. Unless otherwise indicated, all vector and tensor products in symbolic form are assumed to be inner products, such as  $(\mathbf{F}^T \mathbf{F})_{IJ} = F_{iI} F_{iJ}$ , and  $\nabla_X \mathbf{v} : \mathbf{P} = (\partial v_i / \partial X_I) P_{iI}$ , where repeated indices denote a sum over those indices. Uppercase letters are for the most part reserved for variables in the reference configuration, and lowercase letters for the most part designate variables in the current configuration. Cartesian coordinates are assumed. This convention is applied to indices as well:  $S_{IJ}$  is in the reference configuration  $\Omega_0$ , and  $\sigma_{ij}$  in the current configuration  $\Omega$ . 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  $\delta_{IJ}$  is the Kronecker delta operator.

## INTRODUCTION

In many applications related to soft materials, such as soft bending actuators in robotics, accurate prediction of material behavior undergoing large deformation is challenging due to material hyperelasticity and large bending motion [1]. Soft bending actuators are constructed from polymeric or a combination of elastomeric (hyperelastic silicones) and inextensible materials (fabrics and fibers) [2, 3]. In order to characterize the operation of these actuators, experimental, analytical, and numerical approaches are undertaken to describe the behavior of the actuators in terms of geometrical and material parameters. The Finite Element Method (FEM) is one such numerical tool to model these problems computationally, whereby two common element shapes used in three dimensional finite element meshing are tetrahedral and hexahedral elements. In several studies, as an industry-standard FEM software package, ABAQUS [4] has been used to determine relative performance and convergence behavior of tetrahedral and hexahedral elements when combined with material and geometric nonlinearities such as material incompressibility, large deformation and frictional contact [5, 6, 7]. The computational cost of quadratic hexahedral elements is much higher than those of linear hexahedral and tetrahedral elements when they reasonably predict the solution [6]. This is due to the high computational and memory costs and slow solve times when a Jacobian based on quadratic finite elements is assembled [8]. Unless higher accuracy in the numerical solution is required, such elements are avoided in industrial computation. On the other hand, when modeling incompressible (isochoric) or nearly-incompressible hyperelastic materials, due to stabil-

ity, quadratic finite elements ( $Q_2$ ) may be employed since linear hexahedral elements ( $Q_1$ ) may lead to volumetric locking unless enhanced, precluding any meaningful numerical results [9]. Incompressibility may be enforced by satisfying the Babuška-Brezzi condition for a mixed/hybrid formulation that involves interpolating higher-order displacement and lower-order pressure field spaces [10, 11].

To avoid high cost of quadratic elements and the usage of stabilization techniques [12], various element technologies have been developed to address volumetric locking for incompressible and nearly-incompressible materials [12]. Some of these methods employ a low-order displacement field, and a low-order pressure field that may or may not be condensed out at the element level. In many cases, these techniques have been successfully applied to linear elasticity and hyperelasticity at small and large strains [12, 13]. Other techniques for nearly incompressible materials in displacement-only formulations have been used in the literature, such as the penalty method [14, 15, 16], which has also been applied to linearly elastic [17] and viscoelastic materials [18]. For large deformations, a multiplicative decomposition of the deformation gradient into isochoric and volumetric parts is devised that may perform poorly in certain situations [19, 20, 21]. At small strains, Selective Reduced Integration (SRI) is applied separately to the deviatoric and volumetric parts of the stiffness matrix [22, 23]. Due to simplicity of the SRI method, an extension was introduced to handle large deformation elastoplasticity, using an additive split of stress tensor and modulus tensor [9]. However, this method is problem dependent and thus lacks generality. Due to the high cost of higher-order elements such as quadratic elements, linear elements are often employed, and may suffer from stability issues, unless enhanced in some fashion. To overcome the stability issues, as an alternative to the above-mentioned linear elements, it would be desirable to employ a higher-order element at roughly the cost of the linear element while utilizing a pressure field to overcome the locking problem as Poisson's ratio approaches 0.5 for the nearly-incompressible case. In this work, we are interested in the implementation of a displacement-only formulation based on a mixed energy function with a pressure-like variable for a nearly-incompressible Neo-Hookean hyperelastic material deformed at finite strain, with parallel computation applied to unstructured meshes. Reduced integration is considered on the centroid of each element, while higher-order quadrature space is employed for the element. For faster convergence, a matrix-free approach [8] using p-multigrid [24] is applied, with Algebraic Multigrid as the coarse solve on the low-order  $Q_1$  sub-elements. We attempt to numerically determine the limit of attainable near-incompressibility, while enabling fast and parallel computations via the PETSc library [25]. In this method, a higher-order hexahedral element discretization is employed while the basis computation is efficiently performed using a tensor contractions with the matrix-free finite element library libCEED [26, 27, 28, 29],

which provides portable performance with architecture specific implementations across a variety of computational devices.

The remainder of this paper is organized as follows: In sections §1 and §2 we describe the problem's weak form and its discretization approach using a hybrid formulation to implement a Total Lagrangian Neo-Hookean hyperelasticity residual and Jacobian evaluation. In section §3, the details of a preconditioning technique used to accelerate convergence is discussed. The action of the Jacobian operators is applied in a matrix-free format, while a sparse coarse operator, based on linear  $Q_1$  sub-elements on the nodes of the higher-order elements, is assembled. Algebraic Multigrid (AMG) is applied to the assembled matrix in the coarse-solve. In section §4, we discuss our findings based on comparison of our results with the ABAQUS software results.

## 1 Weak Form for Nearly Incompressible Neo-Hookean Hyperelasticity

We consider a mixed strain energy function,

$$\Phi(\mathbf{E}, p) = \mu (\text{tr}(\mathbf{E}) - \log(J)) + p \log(J) - \frac{p^2}{2\lambda}, \quad (1)$$

with  $p(\mathbf{E}) = \lambda \log(J)$  and  $J = \det(\mathbf{F})$ , where  $\mathbf{F}$  is the deformation gradient,  $\log$  is the natural logarithm, and  $\lambda$  and  $\mu$  are the Lamé parameters. For a Neo-Hookean hyperelasticity material we have,  $\mathbf{P} = \mathbf{F}\mathbf{S}$ , where  $\mathbf{P}$  and  $\mathbf{S}$  are the first and second Piola-Kirchhoff stress tensors, respectively.  $\mathbf{S} = 2\mu\mathbf{C}^{-1}\mathbf{E} + p\mathbf{C}^{-1}$  with  $\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I}) = \frac{1}{2}(\nabla_X \mathbf{u} + \nabla_X \mathbf{u}^T + (\nabla_X \mathbf{u})^T (\nabla_X \mathbf{u}))$ , which yields

$$\mathbf{S} = \mu(\mathbf{I} - \mathbf{C}^{-1}) + p\mathbf{C}^{-1}. \quad (2)$$

The strong form of the governing equations for nearly-incompressible Neo-Hookean hyperelastic materials is given by the following system:

$$\begin{cases} \nabla_X \cdot \mathbf{P} = \mathbf{0}, \\ \mathbf{u} = \bar{\mathbf{u}}, \\ \mathbf{P} \cdot \mathbf{N} = \bar{\mathbf{t}}, \\ p - \lambda \log(J) = 0, \end{cases} \quad (3)$$

where  $\nabla_X$  denotes spatial derivative with respect to the reference configuration,  $\rho_0$  is the initial mass density,  $\mathbf{g}$  is the body force per unit mass (such as gravitational acceleration),  $\bar{\mathbf{u}}$  is the essential boundary condition,  $\bar{\mathbf{t}}$  is the prescribed traction (equal to  $\mathbf{0}$  in this work) on the reference configuration, and  $p$  is the pressure field. Introducing weighting functions  $\mathbf{v}$  and  $q$  for displacement and pressure fields, respectively, the corresponding

variational equations (part of the weak form) (3) are given by

$$\int_{\Omega_0} \nabla_X \mathbf{v} : \mathbf{P} dV - \int_{\Omega_0} \mathbf{v} \cdot \rho_0 \mathbf{g} dV - \int_{\partial\Omega_0} \mathbf{v} \cdot \bar{\mathbf{t}} dS = 0, \quad (4)$$

$$\int_{\Omega_0} q \left( \log(J) - \frac{p}{\lambda} \right) dV = 0. \quad (5)$$

We split  $\mathbf{S}$  in equation (2) to account for near-incompressibility of 3D Neo-Hookean hyperelasticity at finite strain where  $\mu(\mathbf{I} - \mathbf{C}^{-1})$  is evaluated using a full quadrature rule, and  $p\mathbf{C}^{-1}$  is evaluated at the centroid of the element. The incremental derivative of the first-Piola Kirchhoff stress using the constitutive equation (2) is derived as,

$$d\mathbf{P} = \frac{\partial \mathbf{P}}{\partial \mathbf{F}} : d\mathbf{F} = d\mathbf{F}\mathbf{S} + \mathbf{F} \underbrace{\frac{\partial \mathbf{S}}{\partial \mathbf{E}} : d\mathbf{E}}_{d\mathbf{S}}, \quad (6)$$

where  $d\mathbf{E} = \frac{\partial \mathbf{E}}{\partial \mathbf{F}} : d\mathbf{F} = \frac{1}{2}(d\mathbf{F}^T \mathbf{F} + \mathbf{F}^T d\mathbf{F})$  and,

$$d\mathbf{S} = -\mu d\mathbf{C}^{-1} + dp\mathbf{C}^{-1} + p d\mathbf{C}^{-1}. \quad (7)$$

The  $dp\mathbf{C}^{-1} + p d\mathbf{C}^{-1}$  terms in equation (7) are evaluated at the centroid of the element, while  $-\mu d\mathbf{C}^{-1}$  is evaluated with the full quadrature rule; we have  $d\mathbf{C}^{-1} = -2\mathbf{C}^{-1} d\mathbf{E}\mathbf{C}^{-1}$  and  $dp = \lambda \mathbf{C}^{-1} : d\mathbf{E}$  since  $p = \lambda \log(J)$ .

## 2 Residual and Action of Jacobian Evaluation

We are interested in the matrix-free implementation of a nearly incompressible Neo-Hookean hyperelasticity model in 3D using continuous global displacement degrees of freedom (dofs) with element-level condensed discontinuous pressure. We employ a mathematical formulation that exposes opportunities for efficient basis operator application via tensor contractions from [8, 30]:

$$\langle \mathbf{v}, \mathbf{f}(\mathbf{u}) \rangle = \int_{\Omega_0} [\mathbf{v} \cdot \mathbf{f}_0(\mathbf{u}, \nabla_X \mathbf{u}) + \nabla_X \mathbf{v} : \mathbf{f}_1(\mathbf{u}, \nabla_X \mathbf{u})] dV = 0. \quad (8)$$

For Neo-Hookean hyperelasticity, we have  $\mathbf{f}_0 = \rho_0 \mathbf{g}$  and  $\mathbf{f}_1 = \mathbf{P}$  from comparing equations (4), (8), where the natural boundary

condition for traction is considered. Additionally, the Jacobian evaluation is given by

$$\begin{aligned} & \langle \mathbf{v}, \mathbf{J}(\mathbf{u})\mathbf{w} \rangle \\ &= \int_{\Omega_0} [\mathbf{v}^T \nabla_X \mathbf{v}^T] \begin{bmatrix} \mathbf{f}_{0,0} & \mathbf{f}_{0,1} \\ \mathbf{f}_{1,0} & \mathbf{f}_{1,1} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \nabla_X \mathbf{w} \end{bmatrix} dV, \end{aligned} \quad (9)$$

with

$$\begin{bmatrix} \mathbf{f}_{0,0} & \mathbf{f}_{0,1} \\ \mathbf{f}_{1,0} & \mathbf{f}_{1,1} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{f}_0}{\partial \mathbf{u}} & \frac{\partial \mathbf{f}_0}{\partial (\nabla_X \mathbf{u})} \\ \frac{\partial \mathbf{f}_1}{\partial \mathbf{u}} & \frac{\partial \mathbf{f}_1}{\partial (\nabla_X \mathbf{u})} \end{bmatrix}. \quad (10)$$

In equation (4),  $\mathbf{f}_0$  is not a function of  $\mathbf{u}$  or  $\nabla_X \mathbf{u}$ . Therefore, its derivative with respect to  $\mathbf{u}$  or  $\nabla_X \mathbf{u}$  is zero, (i.e.,  $\partial \mathbf{f}_0 / \partial \mathbf{u} = \mathbf{0}$  and  $\partial \mathbf{f}_0 / \partial (\nabla_X \mathbf{u}) = \mathbf{0}$ ). On the other hand,  $\mathbf{f}_1$  is a function of  $\nabla_X \mathbf{u}$ , but it is not a function of  $\mathbf{u}$ . Therefore,  $\partial \mathbf{f}_1 / \partial \mathbf{u} = \mathbf{0}$  and  $\partial \mathbf{f}_1 / \partial (\nabla_X \mathbf{u}) = \partial \mathbf{P} / \partial \mathbf{F}$ , whose expression is provided in equation (6).

### 3 Preconditioning with p-multigrid

We compute a matrix-free action of the Jacobian and the residual for higher polynomial degree by employing the tensor contraction algorithms mentioned in the previous section. Since we consider unstructured meshes, Algebraic Multigrid (AMG) is an attractive preconditioning technique which offers fast convergence to solution. However, preconditioning a higher order operator with AMG is inefficient since the assembly of the matrices required by AMG is expensive [24, 31]. Therefore, we utilize the p-multigrid preconditioning technique suggested by [8]. In this method, we reduce high-order elements to  $Q_1$  sub-elements and apply AMG to an assembled Jacobian on the  $Q_1$  sub-elements as a coarse level solve. The prolongation operator is given by evaluating the coarse element Legendre basis polynomials on the fine element Gauss-Legendre-Lobatto (GLL) nodal points, and restriction is given by the transpose of prolongation. In multigrid preconditioning, the smoother is an important component that helps resolve the error correction on a given multigrid level. We use a Chebyshev smoother [32] utilizing the true operator diagonal. Figure 1 represents a  $Q_2 - Q_1$  dual-ordering multigrid step. In what follows, we introduce some of the steps and their operator nomenclature used in our multigrid algorithm.

Prolongation operator:

$$\mathbf{P} = \left( \frac{1}{\mathbf{W}} \right) \sum_e \mathcal{E}_e^T \mathbf{N} \bar{\mathcal{E}}_e,$$

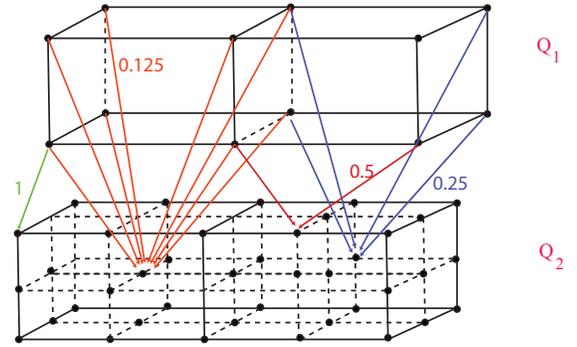
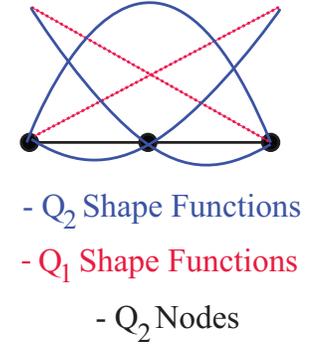


FIGURE 1.  $Q_1$  Basis function evaluation at  $Q_2$  GLL interpolating points, with a  $Q_2 - Q_1$  dual-ordering multigrid step.

and,

$\bar{\mathcal{E}}_e$ : Restrict from global  $Q_1$

$\mathbf{N}$ : Evaluate on  $Q_2$  discretization

$\mathcal{E}_e^T$ : Transpose of element restriction from *fine* discretization

$\mathbf{P}$ : over-counts *face*, *edge* and *vertex* dofs

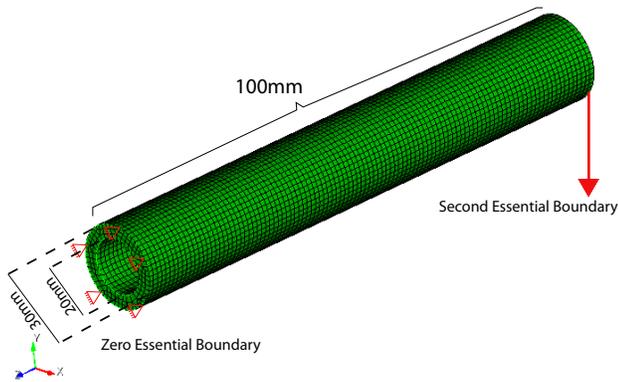
$\mathbf{W}$ : scale node multiplicity:

$$\mathbf{W} = \sum_e \mathcal{E}_e^T \mathbf{1} \quad \left( \frac{1}{\mathbf{W}} \text{ scales each row of } \mathbf{P} \right)$$

Restriction operator:  $\mathbf{P}^T$ .

### 4 Numerical Results

In this section, we provide the numerical results. We consider a Neo-Hookean 3D hyperelasticity model within the static balance of linear-momentum equations mapped back to the reference configuration  $\Omega_0$  for Total Lagrangian FEM in equation (3) where  $\bar{\mathbf{u}}$  is the essential boundary condition. Figure 2 represents a coarse mesh as a model for an assistive finger device generated by the Trellis [33] software (Cubit mesh generator). On the left end we apply a zero Dirichlet boundary condition and on



**FIGURE 2.** Hexahedral mesh for cylindrical tube bending problem. The length of the tube is 100mm, with circular cross-section with inner diameter 10mm and outer diameter 15mm. The left wall is fixed, while the right wall is displaced in the negative  $Y$  direction by 50mm.

the right a non-homogeneous Dirichlet boundary condition in the negative  $Y$  direction, to simulate a tube bending problem.

Poisson's ratio  $\nu = 0.49$  and Young's Modulus  $E = 1$  MPa are considered. The discretized system of equations formulated in sections 1 and 2 are implemented by employing two open source software packages, PETSc version 3.13 and libCEED version 0.6. To leverage the distributed memory environment of parallel computing systems, we rely on PETSc for the domain decomposition and distribution of the dofs across all MPI processes, with each process containing a non-overlapping subset of the entire mesh. PETSc was configured with Exodus-II [34] to handle unstructured meshes. In addition, PETSc is used for parallelization of assembly operations and linear solve (by Conjugate-Gradient (CG)) in each iteration of the Newton solve. libCEED is employed to perform vectorized tensor-product operations over a batch of 8 elements (AVX-2) on each local processor. The discrete Jacobian operator in equation (9) is performed matrix-free, while in the preconditioning with AMG, PETSc's finite difference with coloring is employed for an assembled matrix based on  $Q_1$  elements. AMG is accessed through PETSc's multigrid common interface. For the results showed in Table 1, we use the following stopping parameters and notation:  $10^{-8}$  is the relative stopping condition for Newton-Raphson, "Elem" denotes the number of elements in each mesh, "Nodes" represents the number of nodes in each mesh, "dof" shows the total number of degrees of freedom per mesh, "Loads" represents total number of load increments applied, "Time" is the total CPU compute time in seconds, and "np" represents the number of processors employed for the computation. We have implemented the nearly incompressible Neo-Hookean hyperelastic model into ABAQUS via the user-subroutine UHYPER, since ABAQUS does not allow a user to intentionally turn on finite strain nearly-incompressible

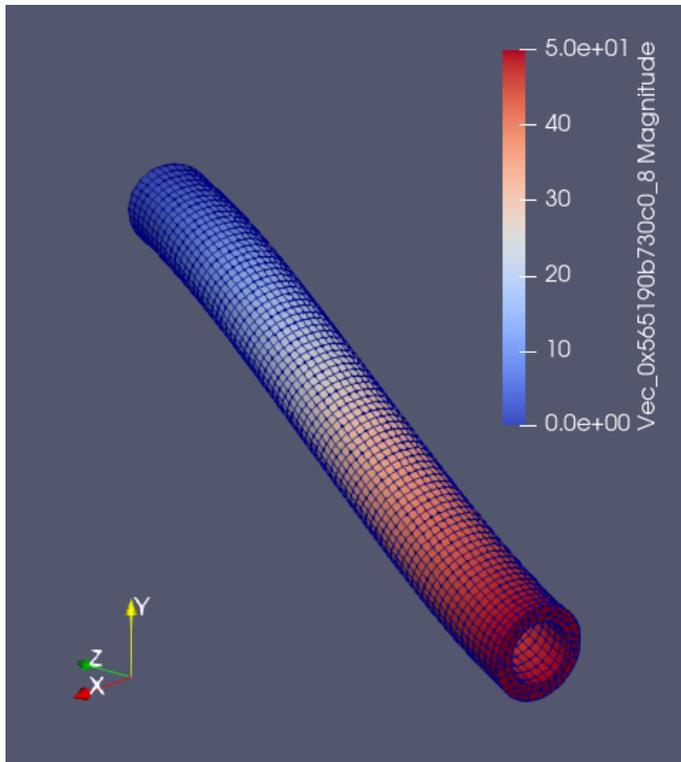
or incompressible hyperelasticity using standard linear, or even quadratic, element types; it requires the Hybrid (or other enhanced) feature enabled for the element type (which is similar to the  $Q_1P_0$  element in [19] and  $Q_2P_1$ ). In order to enable nearly-incompressible hyperelasticity, we introduce a small parameter, denoted by  $\varepsilon$ , such that in the UHYPER subroutine, the Poisson's ratio is  $\nu = 0.5 - \varepsilon$ . In this way, we can control the regimes of near-incompressibility, and investigate how the hybrid built-in element types in ABAQUS perform. Recall that  $\lambda = 2\mu\nu/(1 - 2\nu) \rightarrow \infty$  as  $\nu \rightarrow 0.5$ ; with  $\mu$  the shear modulus.

Two different mesh sizes were used with ABAQUS and PETSc. ABAQUS uses the 20-noded serendipity 3D hexahedral element for displacement interpolation and linear discontinuous pressure, while PETSc imports unstructured hexahedral meshes as 8-noded tri-linear hexahedral element meshes, and internally interpolates the mesh into higher-order hexahedral elements (27-noded tri-quadratic ones in this case). Therefore, the number of elements in Table 1 for PETSc is the number of 8-noded tri-linear hexahedral elements generated by Trelis software. In our PETSc implementation we used constant discontinuous pressure without any corresponding global degree of freedom. Table 1 summarizes a comparison between PETSc and ABAQUS for total CPU time with 8 CPUs. Similar dof and Loads are simulated in each code, whereas in ABAQUS also the 10 Loads case is presented to show that for smaller dof problems (such as simulated here), with the direct solver in ABAQUS (only option available for Hybrid elements), and its nonlinear Newton solver algorithms, it can take larger steps (i.e., fewer Loads increments) than the PETSc code can currently take, leading to shorter total CPU time. Figures 3 and 4 show the deformation of the polymeric material for  $\nu = 0.49$  for PETSc and ABAQUS results.

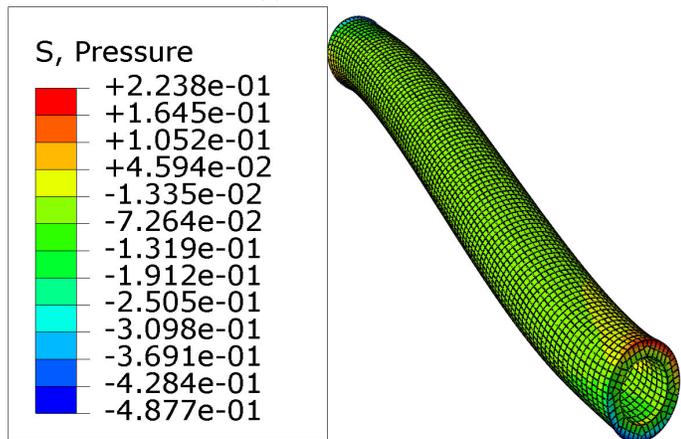
**TABLE 1.** PETSc vs ABAQUS CPU Time with  $\nu = 0.49$

Software	Elem	Nodes	dof	Loads	Time	np
PETSc	736	7,434	22,302	500	474.05	8
	5,550	52,920	158,760	750	8,749.68	8
ABAQUS	1,715	9,945	36,695	500	1500.3	8
	7,896	42,628	159,468	750	12,289.0	8
	7,896	42,628	159,468	10	304.8	8

In the preconditioner, computing with  $\nu$  close to 0.5 introduces numerical instability in the smoothing operators. Therefore, we employ a smaller  $\nu = 0.4$  when computing the operator diagonal for the smoother to avoid this numerical instability. The computation was conducted on a standalone platform with AMD RYZEN 7 3700X 8-Core 3.6 GHz (4.4 GHz Max Boost) Socket AM4 65W 100-100000071BOX Desktop Processor with



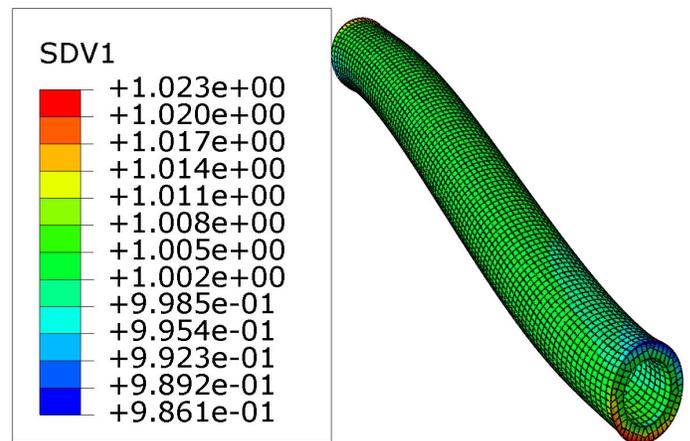
(a) PETSc result.



(b) ABAQUS result.

**FIGURE 3.**  $1\times$  displacement magnitude for deformed mesh for (a) PETSc (displacement magnitude contour in mm), (b) ABAQUS (Cauchy pressure contour in MPa).

G.SKILL Ripjaws V Series 32GB (2 x 16GB) 288-Pin DDR4 SDRAM DDR4 3200 (PC4 25600) Desktop Memory Model F4-3200C16D-32GV. LibCEED along with LIBXSMM [35] was used to target AVX-2 architectures. In the case of polynomial degree 2, quadratic convergence were observed in the Newton



**FIGURE 4.**  $1\times$  displacement magnitude for deformed mesh with  $J = \det \mathbf{F}$  contour (labeled SDV1 in the figure) for ABAQUS. The deformation is nearly-incompressible even with  $\nu = 0.49$ .

solve. However, as the number of degrees of freedom increases, more load increments are required. ABAQUS simulations were conducted on the Blanca Condo Cluster (each node is 2x Intel Xeon Gold 6130 (2.1 GHZ, 16-core, “Skylake”)), administered by Research Computing at the University of Colorado Boulder.

## 5 CONCLUSIONS

The proposed algorithm with condensed-out local pressure field has the advantage of being implemented in parallel and therefore it can be extended to large number of CPUs. In addition, higher accuracy may be expected with polynomial degrees of order 2 and 4 in the solution. However, it performs poorly in the nearly-incompressible case. Under-integration of the local pressure field causes a loss of order of accuracy in each linear solve; therefore, many iterations for the linear solve at each iteration of a Newton solve is required which translate to having many load increments to avoid numerous linear solves. As a result, the time to solution becomes long even though performance is gained via efficient basis operations with tensor-contractions. Also, the limit of the Poisson’s ratio  $\nu \rightarrow 0.5$  challenges the convergence due to the different treatment of the pressure field. This work is considered as a preliminary study of the qualitative behavior of treating near-incompressibility and incompressibility at the centroid of each element when a local pressure field is employed as opposed to a global pressure field. The preconditioner applied to the displacement along with under-integrated pressure field causes poor performance. As a result, for future work, we propose a modification of this algorithm to employ global degrees of freedom for the pressure field with full integration scheme. The resulting system of equations will be a block  $2 \times 2$  matrix where preconditioning the displacement and pres-

sure fields using different approaches is desired for optimal performance.

## REFERENCES

- [1] Wang, Z., Polygerinos, P., Overvelde, J. T., Galloway, K. C., Bertoldi, K., and Walsh, C. J., 2016. “Interaction forces of soft fiber reinforced bending actuators”. *IEEE/ASME Transactions on Mechatronics*, **22**(2), pp. 717–727.
- [2] Galloway, K. C., Polygerinos, P., Walsh, C. J., and Wood, R. J., 2013. “Mechanically programmable bend radius for fiber-reinforced soft actuators”. In 2013 16th International Conference on Advanced Robotics (ICAR), IEEE, pp. 1–6.
- [3] Polygerinos, P., Lyne, S., Wang, Z., Nicolini, L. F., Mosadegh, B., Whitesides, G. M., and Walsh, C. J., 2013. “Towards a soft pneumatic glove for hand rehabilitation”. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp. 1512–1517.
- [4] <https://www.3ds.com/products-services/simulia/products/abaqus/>.
- [5] Tadepalli, S. C., Erdemir, A., and Cavanagh, P. R., 2011. “Comparison of hexahedral and tetrahedral elements in finite element analysis of the foot and footwear”. *Journal of biomechanics*, **44**(12), pp. 2337–2343.
- [6] Benzley, S. E., Perry, E., Merkley, K., Clark, B., and Sjaardama, G., 1995. “A comparison of all hexagonal and all tetrahedral finite element meshes for elastic and elastoplastic analysis”. In Proceedings, 4th international meshing roundtable, Vol. 17, Sandia National Laboratories Albuquerque, NM, pp. 179–191.
- [7] Cifuentes, A., and Kalbag, A., 1992. “A performance study of tetrahedral and hexahedral elements in 3-D finite element structural analysis”. *Finite Elements in Analysis and Design*, **12**(3-4), pp. 313–318.
- [8] Brown, J., 2010. “Efficient nonlinear solvers for nodal high-order finite elements in 3D”. *Journal of Scientific Computing*, **45**(1-3), pp. 48–63.
- [9] Doll, S., Schweizerhof, K., Hauptmann, R., and Freischläger, C., 2000. “On volumetric locking of low-order solid and solid-shell elements for finite elastoviscoplastic deformations and selective reduced integration”. *Engineering Computations*, **17**(7), pp. 874–902.
- [10] Brezzi, F., and Fortin, M., 2012. *Mixed and hybrid finite element methods*, Vol. 15. Springer Science & Business Media.
- [11] Srinivasan, K., Matouš, K., and Geubelle, P. H., 2008. “Generalized finite element method for modeling nearly incompressible bimaterial hyperelastic solids”. *Computer Methods in Applied Mechanics and Engineering*, **197**(51-52), pp. 4882–4893.
- [12] Areias, P., and Matouš, K., 2008. “Stabilized four-node tetrahedron with nonlocal pressure for modeling hyperelastic materials”. *International journal for numerical methods in engineering*, **76**(8), pp. 1185–1201.
- [13] Masud, A., and Xia, K., 2005. “A stabilized mixed finite element method for nearly incompressible elasticity”. *Journal of Applied Mechanics*, **72**(5), pp. 711–720.
- [14] Hughes, T. J. R., 2000. *The finite element method: linear static and dynamic finite element analysis*. Dover Publications Inc., New York.
- [15] der Zanden, J. V., Kuiken, G. D. C., Segal, A., Lindhout, W. J., and Hulsen, M. A., 1985. “Numerical experiments and theoretical analysis of the flow of an elastic liquid of the upper-convected maxwell type in the presence of geometrical discontinuities”. *Appl. Sci. Res.*, **42**, pp. 303–318.
- [16] Zienkiewicz, O. C., Taylor, R. L., and Zhu, J. Z., 2013. *The finite element method: its basis and fundamentals*. Elsevier, Oxford.
- [17] Taylor, R. L., Oñate, E., and Ubach, P.-A., 2005. *Finite Element Analysis of Membrane Structures*. Springer Netherlands, Dordrecht, pp. 47–68.
- [18] Barra, V., Chester, S. A., and Afkhami, S. “Numerical Simulations of Nearly Incompressible Viscoelastic Membranes”. *Computers & Fluids*, **175**, pp. 36–47 (2018).
- [19] Simo, J.-C., Taylor, R. L., and Pister, K., 1985. “Variational and projection methods for the volume constraint in finite deformation elasto-plasticity”. *Computer methods in applied mechanics and engineering*, **51**(1-3), pp. 177–208.
- [20] Simo, J.-C., and Armero, F., 1992. “Geometrically nonlinear enhanced strain mixed methods and the method of incompatible modes”. *International Journal for Numerical Methods in Engineering*, **33**(7), pp. 1413–1449.
- [21] Wriggers, P., and Reese, S., 1996. “A note on enhanced strain methods for large deformations”. *Computer Methods in Applied Mechanics and Engineering*, **135**(3-4), pp. 201–209.
- [22] Liu, C., Hofstetter, G., and Mang, H., 1992. “Evaluation of 3d FE-formulations for incompressible hyperelastic materials at finite strains”. In Proceedings of the First European Conference on Numerical Methods in Engineering, Vol. 7, Sept.
- [23] Liu, C., Hofstetter, G., and Mang, H., 1994. “3D finite element analysis of rubber-like materials at finite strains”. *Engineering computations*, **11**(2), pp. 111–128.
- [24] Sundar, H., Stadler, G., and Biros, G., 2015. “Comparison of multigrid algorithms for high-order continuous finite element discretizations”. *Numerical Linear Algebra with Applications*, **22**(4), pp. 664–680.
- [25] Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W. D., Kaushik, D., Knepley, M. G., May, D. A., McInnes, L. C., Mills, R. T., Munson, T., Rupp, K., Sanan, P., Smith, B. F., Zampini, S., Zhang, H., and Zhang, H.,

2018. PETSc Web page. <http://www.mcs.anl.gov/petsc>.
- [26] libCEED development site, 2020. <https://github.com/ceed/libceed>.
- [27] libCEED user manual, 2020. <https://libceed.readthedocs.io/en/latest/>.
- [28] Brown, J., Abdelfata, A., Camier, J.-S., Dobrev, V., Dongarra, J., Fischer, P., Fisher, A., Dudouit, Y., Haidar, A., Kamran, K., Klev, T., Min, M., Ratnayaka, T., Shephard, M., Smith, C., Tomov, S., Tomov, V., and Warburton, T., 2018. “CEED ECP Milestone Report: Public release of CEED 1.0”. Tech. rep.
- [29] Brown, J., Abdelfattah, A., Barra, V., Dobrev, V., Dudouit, Y., Fischer, P., Kolev, T., Medina, D., Min, M., Ratnayaka, T., Smith, C., Thompson, J., Tomov, S., Tomov, V., and Warburton, T., 2019. “CEED ECP Milestone Report: Public release of CEED 2.0”. Tech. rep.
- [30] Knepley, M. G., Brown, J., Rupp, K., and Smith, B. F., 2013. “Achieving high performance with unified residual evaluation”. *arXiv preprint arXiv:1309.1204*.
- [31] Heys, J., Manteuffel, T., McCormick, S., and Olson, L., 2005. “Algebraic multigrid for higher-order finite elements”. *Journal of computational Physics*, **204**(2), pp. 520–532.
- [32] Adams, M., Brezina, M., Hu, J., and Tuminaro, R., 2003. “Parallel multigrid smoothing: polynomial versus Gauss-Seidel”. *Journal of Computational Physics*, **188**(2), pp. 593–610.
- [33] <http://csimsoft.com>.
- [34] <https://gsjaardema.github.io/seacas/html/index.html>.
- [35] <https://github.com/hfp/libxsmm>.