

Three-Dimensional Isogeometric Analysis of Flexoelectricity with MATLAB Implementation

Hamid Ghasemi¹, Harold S. Park², Xiaoying Zhuang^{3,4,*} and Timon Rabczuk^{5,6}

Abstract: Flexoelectricity is a general electromechanical phenomenon where the electric polarization exhibits a linear dependency to the gradient of mechanical strain and vice versa. The truncated pyramid compression test is among the most common setups to estimate the flexoelectric effect. We present a three-dimensional isogeometric formulation of flexoelectricity with its MATLAB implementation for a truncated pyramid setup. Besides educational purposes, this paper presents a precise computational model to illustrate how the localization of strain gradients around pyramidal boundary shapes contributes in generation of electrical energy. The MATLAB code is supposed to help learners in the Isogeometric Analysis and Finite Elements Methods community to learn how to solve a fully coupled problem, which requires higher order approximations, numerically. The complete MATLAB code which is available as source code distributed under a BSD-style license, is provided in the part of Supplementary Materials of the paper.

Keywords: Flexoelectricity, Isogeometric Analysis (IGA), MATLAB, B-spline elements, finite elements, coupled electromechanical problem.

1 Introduction

In dielectric crystals with non-centrosymmetric crystal structure such as quartz and ZnO, electrical polarization is generated upon the application of uniform mechanical strain. This property of certain materials, which is known as piezoelectricity, is caused by relative displacements between the centers of oppositely charged ions. Flexoelectricity which has recently attracted significant attention; however, differs from the piezoelectricity in two aspects: 1) it is a more general effect which is available in all dielectrics including those with centrosymmetric crystal structures and 2) the induced electrical polarization is related

¹ Department of Mechanical Engineering, Arak University of Technology, Arak, 38181-41167, Iran.

² Department of Mechanical Engineering, Boston University, Boston, MA 02215, USA.

³ Division of Computational Mechanics, Ton Duc Thang University, Ho Chi Minh City, Vietnam.

⁴ Faculty of Civil Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam.

⁵ Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia.

⁶ Institute of Structural Mechanics, Bauhaus-Universität Weimar, Weimar, D-99423, Germany.

* Corresponding Author: Xiaoying Zhuang. Email: Xiaoying.zhuang@tdtu.edu.vn.

Received: 15 August 2019; Accepted: 29 August 2019.

to the gradient of mechanical strain and is thus a size dependent effect [Sharma, Maranganti and Sharma (2007); Yudin and Tagantsev (2013)].

The compression of a truncated pyramid is a common setup to measure flexoelectric properties of dielectrics [Zhu, Fu, Li et al. (2006); Abdollahi, Millan, Peco et al. (2015); Huang, Shu, Kwon et al. (2014); Kwon (2017)]. The pyramidal boundary shapes introduce strain gradients and thus induce electrical voltage. By recording these two quantities, one can quantify flexoelectricity [Cross (2006)]. Sometimes the experimental and theoretical measurements of flexoelectric coefficients differ, noticeably [Sharma, Landis and Sharma (2010)]. One potential reason is lack of a precise mathematical model. Available analytical models [Sharma, Landis and Sharma (2010); Maranganti, Sharma and Sharma (2006); Eliseev, Morozovska, Glinchuk et al. (2009); Mao and Purohit (2014)] are not capable to address the complexity of the deformation fields; particularly around the edges of the pyramid. There are also computational models [Qi, Huang, Fu et al. (2018); Shen and Hu (2010); Mao, Ai, Xiang et al. (2016); Yvonnet and Liu (2017); Mao, Purohit and Aravas (2016); Nguyen, Zhuang and Rabczuk (2018)]. Abdollahi et al. [Abdollahi, Peco, Millan et al. (2014); Abdollahi, Peco, Millan et al. (2015)] presented a meshfree model in 2D. Thai et al. [Thai, Rabczuk and Zhuang (2018)] presented a large deformation isogeometric approach for flexoelectricity. He et al. [He, Javvaji and Zhuang (2019)] implemented element-free Galerkin method to characterize flexoelectricity in a composite material. Sidhardh et al. [Sidhardh and Ray (2018)], presented a numerical model to obtain the effective properties of flexoelectric fiber-reinforced nanocomposite. The authors of this paper already presented an IGA model for flexoelectricity in two-dimensional space [Ghasemi, Park and Rabczuk (2017, 2018)]. It is noteworthy to mention that, only a few of available computational models are in 3D [Abdollahi, Millan, Peco et al. (2015); Deng, Deng and Shen (2018); Codony, Marco, Fern'andez-M'endez et al. (2019); Liu, Wang, Xu et al. (2018); Poya, Gil, Ortigosa et al. (2019)]. This is our motivation to extend our previous work into three-dimensional space for precisely obtain the mechanical strains near the edges of the pyramid.

Proposed by Hughes and his co-workers [Hughes, Cottrell and Bazilevs (2005)], the basic idea behind IGA was to unify Computer Aided Design (CAD) and Computer Aided Engineering (CAE). IGA shows some advantages in comparison to the classical FEM. Among them one can point to the exact representation of the geometry, ease in adaptivity and mesh refinement, accuracy in imposing the essential boundary conditions and the higher and controllable continuity at the inter-element boundary. Here, IGA benefits us with compact support high order B-spline basis functions to discretize the fourth order partial differential equations of flexoelectricity; which demand at least C^1 continuous basis functions in Galerkin method [Abdollahi, Millan, Arroyo et al. (2014)].

We present a MATLAB implementation for a 3D isogeometric formulation of flexoelectricity. We provide necessary tutorials for each section of the code, making it straightforward to follow. Hopefully, it will contribute to the popularity of the topic. The remainder of this paper is organized thus: Section 2 summarizes the discretized governing equations of flexoelectricity, Section 3 explains MATLAB implementation, Section 4 discusses about results and Section 5 offers concluding remarks.

2 A summary of the governing equations and discretization

2.1 Governing equations

A summary of the governing equations of the flexoelectricity is presented in this section. More details are available in [Abdollahi, Peco, Millan et al. (2014); Ghasemi, Park and Rabczuk (2017)] and references therein. Accounting for the flexoelectricity, the enthalpy density, \mathcal{H} , can be written as

$$\mathcal{H}(\varepsilon_{ij}, E_i, \varepsilon_{jk,l}) = \frac{1}{2} C_{ijkl} \varepsilon_{ij} \varepsilon_{kl} - \mu_{ijkl} E_i \varepsilon_{jk,l} - \frac{1}{2} \kappa_{ij} E_i E_j + \frac{1}{2} h_{ijklmn} \varepsilon_{ij,k} \varepsilon_{lm,n} \quad (1)$$

where C_{ijkl} is the fourth-order elasticity tensor, ε_{ij} is the mechanical strain, E_i is the electric field, μ_{ijkl} is the fourth-order flexoelectric tensor, h_{ijklmn} is the sixth-order strain-gradient elasticity tensor and κ_{ij} is the second-order dielectric tensor.

The different stresses / electric displacements including the usual ($\hat{\sigma}_{ij} / \hat{D}_i$), higher-order ($\tilde{\sigma}_{ijk} / \tilde{D}_{ij}$) and physical (σ_{ij} / D_i) ones are then defined through the following relations

$$\hat{\sigma}_{ij} = \frac{\partial \mathcal{H}}{\partial \varepsilon_{ij}} \quad \text{and} \quad \hat{D}_i = -\frac{\partial \mathcal{H}}{\partial E_i} \quad (2)$$

$$\tilde{\sigma}_{ijk} = \frac{\partial \mathcal{H}}{\partial \varepsilon_{ij,k}} \quad \text{and} \quad \tilde{D}_{ij} = -\frac{\partial \mathcal{H}}{\partial E_{i,j}} \quad (3)$$

$$\sigma_{ij} = \hat{\sigma}_{ij} - \tilde{\sigma}_{ijk,k} \quad \text{and} \quad D_i = \hat{D}_i - \tilde{D}_{ij,j} \quad (4)$$

thus

$$\sigma_{ij} = \hat{\sigma}_{ij} - \tilde{\sigma}_{ijk,k} = C_{ijkl} \varepsilon_{kl} + \mu_{lijk} E_{l,k} - h_{ijklmn} \varepsilon_{lm,nk} \quad (5)$$

$$D_i = \hat{D}_i - \tilde{D}_{ij,j} = \kappa_{ij} E_j + \mu_{klj} \varepsilon_{kl,j} \quad (6)$$

which are the governing equations of the flexoelectricity. By imposing boundary conditions and integration over the domain, Ω , the total electrical enthalpy is

$$H = \frac{1}{2} \int_{\Omega} (\hat{\sigma}_{ij} \varepsilon_{ij} + \tilde{\sigma}_{ijk} \varepsilon_{ij,k} - \hat{D}_i E_i) d\Omega \quad (7)$$

Using Hamilton's principle, we finally have

$$\int_{\Omega} (C_{ijkl} \delta \varepsilon_{ij} \varepsilon_{kl} - \mu_{lijk} E_l \delta \varepsilon_{ij,k} - \kappa_{ij} \delta E_i E_j - \mu_{ijkl} \delta E_i \varepsilon_{jk,l} + h_{ijklmn} \delta \varepsilon_{ij,k} \varepsilon_{lm,n}) d\Omega - \int_{\Gamma_t} \bar{t}_i \delta u_i dS + \int_{\Gamma_D} \varpi \delta \theta dS = 0 \quad (8)$$

which is the weak form of the governing equations of the flexoelectricity. In Eq. (8), u_i is the mechanical displacements, θ is the electric potential, \bar{t}_i is the prescribed mechanical tractions and ϖ is the surface charge density. Γ_t and Γ_D are boundaries of Ω corresponding to mechanical tractions and electric displacements, respectively.

2.2 IGA discretization

There are two different spaces in IGA namely the physical space (Fig. 1(c)) and parameter space (Fig. 1(b)). Knot vectors discretize the parameter space. A knot vector in one dimension is a non-decreasing set of coordinates in the parameter space. We call knot vectors corresponding to X , Y and Z directions as $\xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$, $\eta =$

$\{\eta_1, \eta_2, \dots, \eta_{m+q+1}\}$ and $\chi = \{\chi_1, \chi_2, \dots, \chi_{l+r+1}\}$ respectively. i is called the knot index and $\xi_i, \eta_i, \chi_i \in \mathbb{R}$ are the i^{th} knot in each direction. n, m and l are the number of basis functions; while p, q and r are their polynomial orders. The parent element (Fig. 1(a)) is used for numerical integration. If knots are equally-spaced in the parametric space, they are said to be uniform. A knot vector is said to be open if its first and last knots appear $p + 1$ times.

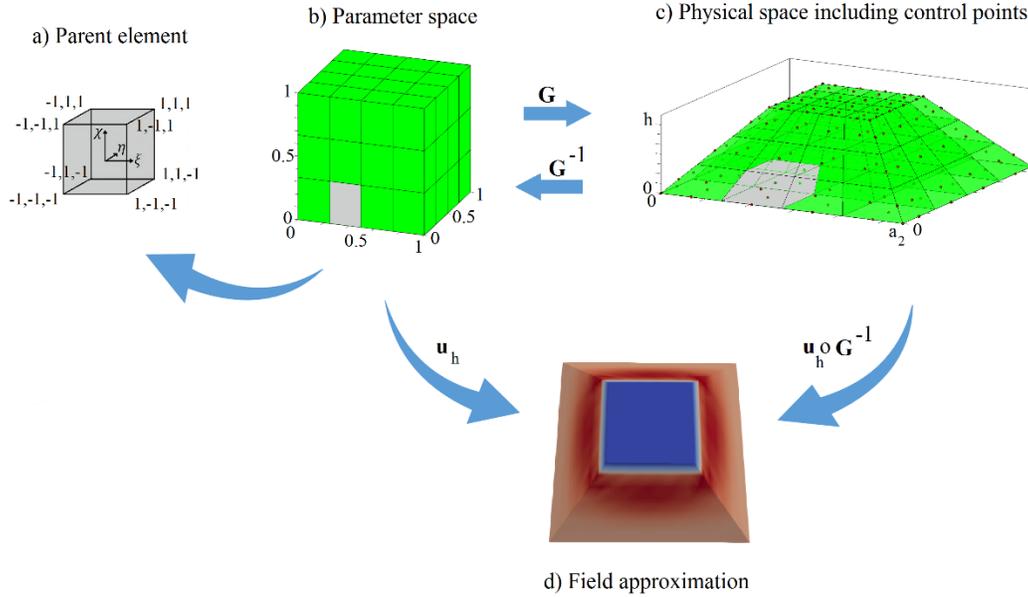


Figure 1: IGA concept: parent element (a), parameter space (b), physical space and control points (c) and field approximation (d). G, G^{-1} and u_h denote mapping, inverse mapping and a solution field, respectively

B-spline basis functions, $N_{i,p}(\xi)$ are recursively defined by using Cox-de Boor formula and starting with piecewise constants ($p = 0$) [Hughes, Cottrell and Bazilevs (2005)]

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{9a}$$

and for $p = 1, 2, 3, \dots$

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{9b}$$

assuming that $\frac{0}{0} = 0$. The derivatives of basis functions, $N'_{i,p}(\xi)$ are given by

$$N'_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{10}$$

Among important B-spline basis functions properties, one can point to partition of unity ($\sum_{i=1}^n N_{i,p}(\xi) = 1$) and non-negativeness ($N_{i,p}(\xi) \geq 0, \forall \xi$) as observable in Fig. 2.

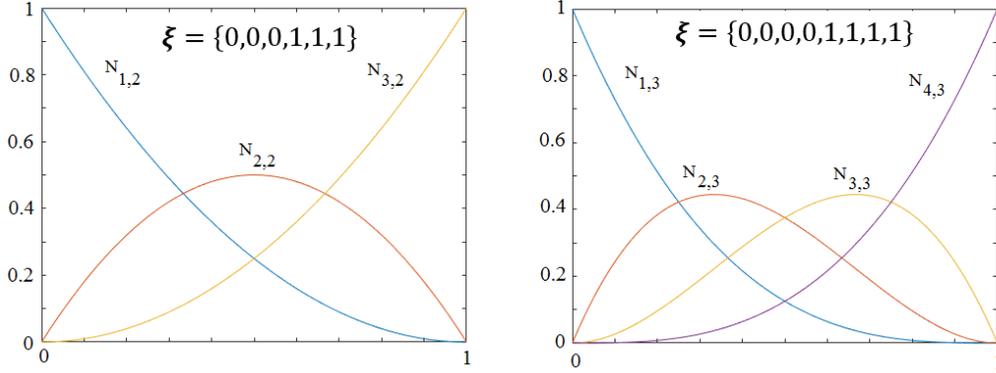


Figure 2: Order elevation, Quadratic ($p = 2$) B-spline basis functions for an open uniform knot vector $\xi = \{0,0,0,1,1,1\}$ (a), and cubic ($p = 3$) B-spline basis functions for an open uniform knot vector $\xi = \{0,0,0,0,1,1,1,1\}$ (b)

Basis functions can be enriched by increasing their order. They are C^{p-m} continuous (i.e., there are $p - m$ continuous derivatives) across element boundaries, where p is polynomial order and m is multiplicity of each knot value. This higher continuity is our main motivation to use B-spline basis functions to discretize the fourth order PDE of flexoelectricity.

Fig. 3 shows that, the support of a B-spline function of order p is $p + 1$ knot spans; it means $N_{i,p}$ is non-zero over $\{\xi_i, \xi_{i+p+1}\}$. There is a notion of k -refinement in IGA, which is an order elevation to p followed by a knot insertion, to obtain C^{p-1} continuity of the basis functions at inserted knot.

Assuming $N_{i,p}(\xi)$, $M_{j,q}(\eta)$ and $P_{k,r}(\chi)$ to be univariate B-spline basis functions of order p , q and r corresponding to knot vector ξ , η and χ , B-spline basis functions in three dimensional space, $N_{i,j,k}^{p,q,r}(\xi, \eta, \chi)$ are presented as

$$N_{i,j,k}^{p,q,r}(\xi, \eta, \chi) = N_{i,p}(\xi) M_{j,q}(\eta) P_{k,r}(\chi) \tag{11a}$$

with the matrix form (for cubic basis functions) as

$$N_{i,j,k}^{p,q,r}(\xi, \eta, \chi) = [P_1 M_1 N_1 \quad P_1 M_1 N_2 \quad P_1 M_1 N_3 \quad P_1 M_2 N_1 \quad P_1 M_2 N_2 \quad \dots \quad P_3 M_3 N_1 \quad P_3 M_3 N_2 \quad P_3 M_3 N_3]^T \tag{11b}$$

Control points (red dots in Fig. 1(c)) in IGA are used to discretize the geometry and define the degrees of freedom. They do not necessarily lie on the surface itself, but define its envelope. As shown in Fig. 1, each element in the physical space is the image of a corresponding element in the parameter space (and vice versa) via mapping \mathbf{G} and \mathbf{G}^{-1} , respectively. Solution fields (e.g., \mathbf{u}_h) can be similarly approximated via tensor product of nodal values (either coordinates or solution fields) with their corresponding basis functions

$$\mathbf{F}(\xi, \eta, \chi) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l N_{i,j,k}^{p,q,r}(\xi, \eta, \chi) \mathbf{Q}_{i,j,k} \tag{12}$$

where $\{\mathbf{Q}_{i,j,k}\}$ is the corresponding control points of the element.

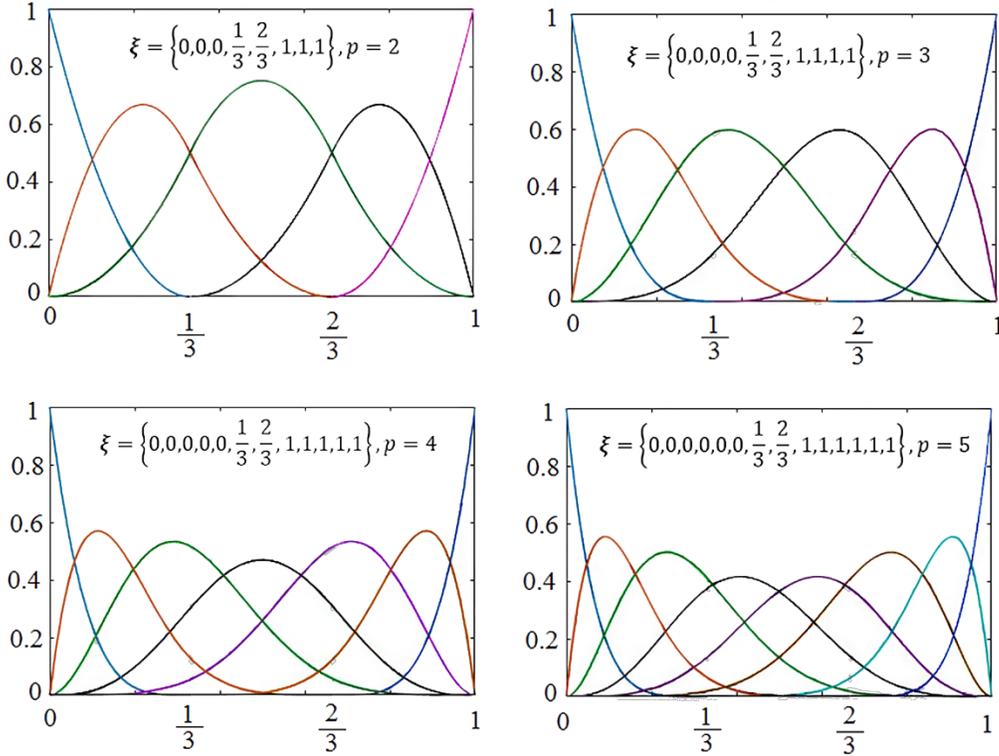


Figure 3: k -refinement of three element, higher-order meshes, each basis-function is C^{p-1} across element boundaries

For displacement \mathbf{u}_h and electric potential θ_h fields, we have

$$\mathbf{u}_h(x, y, z) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l \mathbf{N}_{i,j,k}^{p,q,r}(\xi, \eta, \chi) \mathbf{u}_{i,j,k}^e = (\mathbf{N}_u)^T \mathbf{u}^e \tag{13a}$$

$$\theta_h(x, y, z) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l N_{i,j,k}^{p,q,r}(\xi, \eta, \chi) \theta_{i,j,k}^e = (\mathbf{N}_\theta)^T \boldsymbol{\theta}^e \tag{13b}$$

$$\partial_j u_h = \partial_j (\mathbf{N}_u)^T \mathbf{u}^e = (\mathbf{B}_u)^T \mathbf{u}^e = \boldsymbol{\varepsilon} \tag{13c}$$

$$\partial_j \theta_h = \partial_j (\mathbf{N}_\theta)^T \boldsymbol{\theta}^e = (\mathbf{B}_\theta)^T \boldsymbol{\theta}^e = -\mathbf{E} \tag{13d}$$

$$\partial_j \partial_k u_h = \partial_j \partial_k (\mathbf{N}_u)^T \mathbf{u}^e = (\mathbf{H}_u)^T \mathbf{u}^e \tag{13e}$$

$$\partial_j \partial_k \theta_h = \partial_j \partial_k (\mathbf{N}_\theta)^T \boldsymbol{\theta}^e = (\mathbf{H}_s)^T \boldsymbol{\theta}^e \tag{13f}$$

where the superscript e denotes nodal parameters at the control points;

$$(\mathbf{B}_u)^T = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 & 0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} \\ 0 & \frac{\partial N_i}{\partial y} & 0 & \frac{\partial N_i}{\partial z} & 0 & \frac{\partial N_i}{\partial x} \\ 0 & 0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0 \end{bmatrix} \text{ and } \mathbf{B}_\theta = \begin{bmatrix} \frac{\partial N_i}{\partial x} & \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial z} \\ \vdots & \vdots & \vdots \end{bmatrix} \text{ contain the spatial}$$

derivatives of the B-spline basis functions.

\mathbf{H}_u and \mathbf{H}_s are Hessian matrices containing the second derivatives of the basis functions:

$$\mathbf{H}_u = \begin{bmatrix} \frac{\partial^2 N_i}{\partial x^2} & \frac{\partial^2 N_i}{\partial x \partial y} & \frac{\partial^2 N_i}{\partial x \partial z} & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \frac{\partial^2 N_i}{\partial y \partial x} & \frac{\partial^2 N_i}{\partial y^2} & \frac{\partial^2 N_i}{\partial y \partial z} & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial^2 N_i}{\partial z \partial x} & \frac{\partial^2 N_i}{\partial z \partial y} & \frac{\partial^2 N_i}{\partial z^2} & \dots \\ \dots & 0 & 0 & 0 & \frac{\partial^2 N_i}{\partial z \partial x} & \frac{\partial^2 N_i}{\partial z \partial y} & \frac{\partial^2 N_i}{\partial z^2} & \frac{\partial^2 N_i}{\partial y \partial x} & \frac{\partial^2 N_i}{\partial y^2} & \frac{\partial^2 N_i}{\partial y \partial z} \\ \dots & \frac{\partial^2 N_i}{\partial z \partial x} & \frac{\partial^2 N_i}{\partial z \partial y} & \frac{\partial^2 N_i}{\partial z^2} & 0 & 0 & 0 & \frac{\partial^2 N_i}{\partial x^2} & \frac{\partial^2 N_i}{\partial x \partial y} & \frac{\partial^2 N_i}{\partial x \partial z} \\ \dots & \frac{\partial^2 N_i}{\partial y \partial x} & \frac{\partial^2 N_i}{\partial y^2} & \frac{\partial^2 N_i}{\partial y \partial z} & \frac{\partial^2 N_i}{\partial x^2} & \frac{\partial^2 N_i}{\partial x \partial y} & \frac{\partial^2 N_i}{\partial x \partial z} & 0 & 0 & 0 \end{bmatrix} \text{ and}$$

$$\mathbf{H}_s = \begin{bmatrix} \frac{\partial^2 N_i}{\partial x^2} & 0 & 0 & \frac{\partial^2 N_i}{\partial x \partial y} & 0 & 0 & \frac{\partial^2 N_i}{\partial x \partial z} & 0 & 0 & \dots \\ 0 & \frac{\partial^2 N_i}{\partial y^2} & 0 & 0 & \frac{\partial^2 N_i}{\partial y \partial x} & 0 & 0 & \frac{\partial^2 N_i}{\partial y \partial z} & 0 & \dots \\ 0 & 0 & \frac{\partial^2 N_i}{\partial z^2} & 0 & 0 & \frac{\partial^2 N_i}{\partial z \partial x} & 0 & 0 & \frac{\partial^2 N_i}{\partial z \partial y} & \dots \\ \dots & \frac{\partial^2 N_i}{\partial y \partial x} & \frac{\partial^2 N_i}{\partial z \partial x} & 0 & \frac{\partial^2 N_i}{\partial y^2} & \frac{\partial^2 N_i}{\partial z \partial y} & 0 & \frac{\partial^2 N_i}{\partial y \partial z} & \frac{\partial^2 N_i}{\partial z^2} & 0 \\ \dots & \frac{\partial^2 N_i}{\partial x^2} & 0 & \frac{\partial^2 N_i}{\partial z \partial x} & \frac{\partial^2 N_i}{\partial x \partial y} & 0 & \frac{\partial^2 N_i}{\partial z \partial y} & \frac{\partial^2 N_i}{\partial x \partial z} & 0 & \frac{\partial^2 N_i}{\partial z^2} \\ \dots & 0 & \frac{\partial^2 N_i}{\partial x^2} & \frac{\partial^2 N_i}{\partial y \partial x} & 0 & \frac{\partial^2 N_i}{\partial x \partial y} & \frac{\partial^2 N_i}{\partial y^2} & 0 & \frac{\partial^2 N_i}{\partial x \partial z} & \frac{\partial^2 N_i}{\partial y \partial z} \end{bmatrix}$$

The discrete system of Eq. (8) is expressed as

$$\begin{bmatrix} \mathbf{A}_{UU} & \mathbf{A}_{U\theta} \\ \mathbf{A}_{\theta U} & \mathbf{A}_{\theta\theta} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \theta \end{bmatrix} = \begin{bmatrix} \mathbf{f}_U \\ \mathbf{f}_\theta \end{bmatrix} \tag{14}$$

with

$$\mathbf{A}_{UU} = \sum_e \int_{\Omega_e} [(\mathbf{B}_u)^T \mathbf{C} (\mathbf{B}_u) + (\mathbf{H}_s) \mathbf{h} (\mathbf{H}_s)^T] d\Omega \tag{14a}$$

$$\mathbf{A}_{U\theta} = \sum_e \int_{\Omega_e} (\mathbf{H}_u) \boldsymbol{\mu}^T (\mathbf{B}_\theta)^T d\Omega \tag{14b}$$

$$\mathbf{A}_{\theta U} = \sum_e \int_{\Omega_e} (\mathbf{B}_\theta) \boldsymbol{\mu} (\mathbf{H}_u)^T d\Omega \tag{14c}$$

$$\mathbf{A}_{\theta\theta} = -\sum_e \int_{\Omega_e} (\mathbf{B}_\theta) \boldsymbol{\kappa} (\mathbf{B}_\theta)^T d\Omega \tag{14d}$$

$$\mathbf{f}_U = \sum_e \int_{\Gamma_{t_e}} \mathbf{N}_u^T \mathbf{t}_\Gamma ds \tag{14e}$$

$$\mathbf{f}_\theta = -\sum_e \int_{\Gamma_{D_e}} \mathbf{N}_\theta^T \varpi ds \tag{14f}$$

In Eqs. (14a)-(14f), the subscript, e , in Ω_e , Γ_{t_e} and Γ_{D_e} denotes the e^{th} finite element where $\Omega = \cup_e \Omega_e$. \mathbf{f}_U and \mathbf{f}_θ are mechanical and electrical load vectors. We consider isotropic elasticity and permittivity tensors and adopt cubic symmetry for flexoelectric tensor. Thus, \mathbf{C} , $\boldsymbol{\kappa}$, \mathbf{e} and $\boldsymbol{\mu}$ read as follows:

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & c_{12} & 0 & 0 & 0 \\ c_{12} & c_{11} & c_{12} & 0 & 0 & 0 \\ c_{12} & c_{12} & c_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{44} \end{bmatrix} \tag{15a}$$

with $c_{11} = Y(1 - \nu)/(1 + \nu)(1 - 2\nu)$; $c_{12} = Y\nu/(1 + \nu)(1 - 2\nu)$ and $c_{44} = (c_{11} - c_{12})/2$; where ν denotes Poisson's ratio and Y is the Young's modulus.

$$\boldsymbol{\kappa} = \begin{bmatrix} \kappa_{11} & 0 & 0 \\ 0 & \kappa_{11} & 0 \\ 0 & 0 & \kappa_{11} \end{bmatrix} \tag{15b}$$

,

$$[\boldsymbol{\mu}]_{3 \times 18} = \begin{bmatrix} \mu_{11} & 0 & 0 & \mu_{12} & 0 & 0 & \mu_{12} & 0 & 0 & \dots & \dots & \dots \\ 0 & \mu_{12} & 0 & 0 & \mu_{11} & 0 & 0 & \mu_{12} & 0 & \dots & \dots & \dots \\ 0 & 0 & \mu_{12} & 0 & 0 & \mu_{12} & 0 & 0 & \mu_{11} & \dots & \dots & \dots \\ & & & \dots & 0 & 0 & 0 & 0 & \mu_{44} & 0 & \mu_{44} & 0 \\ & & & \dots & 0 & 0 & \mu_{44} & 0 & 0 & 0 & \mu_{44} & 0 \\ & & & \dots & 0 & \mu_{44} & 0 & \mu_{44} & 0 & 0 & 0 & 0 \end{bmatrix} \tag{15c}$$

with $\mu_{44} = 0$ and

$$[\mathbf{h}]_{18 \times 18} = (l_1)^2 \begin{bmatrix} c_{11} & 0 & 0 & 0 & c_{12} & c_{12} & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & c_{11} & 0 & c_{12} & 0 & 0 & 0 & 0 & c_{12} & \vdots & \vdots & \vdots \\ 0 & 0 & c_{11} & 0 & 0 & 0 & c_{12} & c_{12} & 0 & & & \\ 0 & c_{12} & 0 & c_{11} & 0 & 0 & 0 & 0 & c_{12} & & & \\ c_{12} & 0 & 0 & 0 & c_{11} & c_{12} & 0 & 0 & 0 & & & \\ c_{12} & 0 & 0 & 0 & c_{12} & c_{11} & 0 & 0 & 0 & & & \\ 0 & 0 & c_{12} & 0 & 0 & 0 & c_{11} & c_{12} & 0 & & & \\ 0 & 0 & c_{12} & 0 & 0 & 0 & c_{12} & c_{11} & 0 & & & \\ 0 & c_{12} & 0 & c_{12} & 0 & 0 & 0 & 0 & c_{11} & 0 & & \\ 0 & \dots & & & & & & & 0 & c_{44} & 0 & \\ 0 & \dots & & & & & & & 0 & 0 & \ddots & 0 \\ 0 & \dots & & & & & & & 0 & c_{44} & 0 & \end{bmatrix} \tag{15d}$$

where, l_1 is the length scale to guarantee positive definiteness of the strain energy [Abdollahi, Millan, Peco et al. (2015)].

3 MATLAB implementation

Due to difference in surface areas, the truncated pyramid experiences different tractions on the top and bottom surfaces. This results in strain gradient and consequently, electrical polarization. We consider a pyramid subject to a uniform pressure on its top surface with dimensions and material properties listed in Tab. 1. The boundary conditions are demonstrated in Fig. 4. We use cubic B-spline elements, unless otherwise specified.

Table 1: Dimensions and material properties of the pyramid (barium strontium titanate, BST) [Abdollahi, Millan, Peco et al. (2015)]

a_2/a_1	h	α	ν	$\mu_{11} = \mu_{12}$	κ_{11}	Y	F	l_1
2.72/1.13	0.76	$\pi/4$	0.33	121	141.6	152	200	10 nm
<i>mm</i>	<i>mm</i>			$\mu C/m$	nC/Vm	<i>GPa</i>	<i>N</i>	

Legends: α : inclination angle, Y : Young’s modulus, ν : Poisson’s ratio, l_1 : Length scale, μ_{11}/μ_{12} : flexoelectric constants, κ_{11}/κ_{33} : dielectric constants

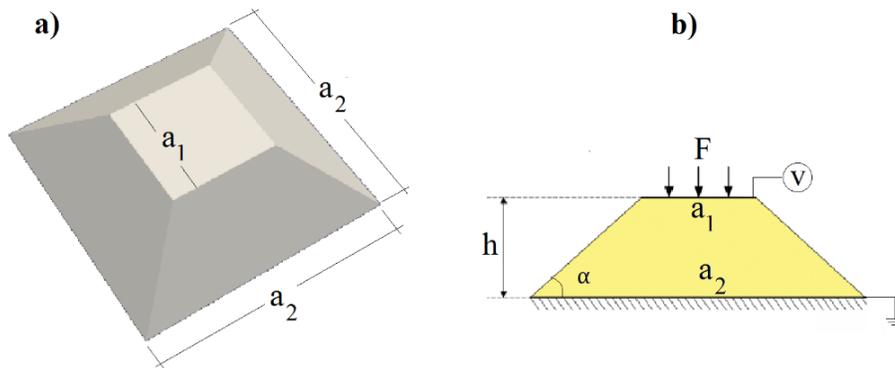


Figure 4: The geometry of the pyramid (a) loading and boundary conditions (b)

3.1 Element structure and connectivity

The function *CUBIC_Mesh* generates cubic elements with eight vertices in the parametric space. The inputs of the function are number of elements in X, Y and Z directions as well as the number of nodes per element which is eight. The function returns the nodes coordinates and connectivity as configured in Fig. 5.

The function *cppolygonPrint_3D_truncated* returns coordinates of the control points. For a typical quadratic element, the corresponding control points are identified according to the pattern shown in Fig. 6. The function *pageomapping_3D_truncated* (Listing 1) maps all elements nodes from the parameter space onto the physical space.

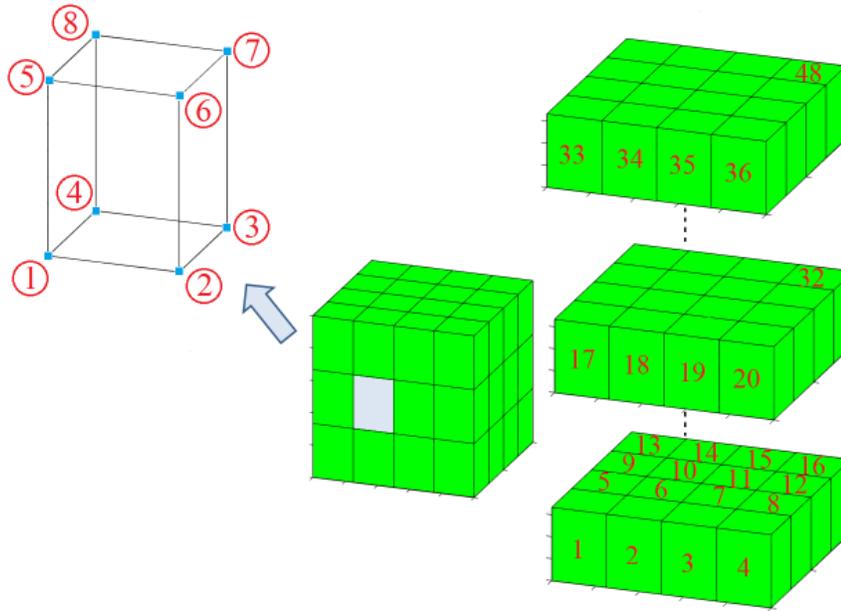


Figure 5: Elements connectivity and local node numbering

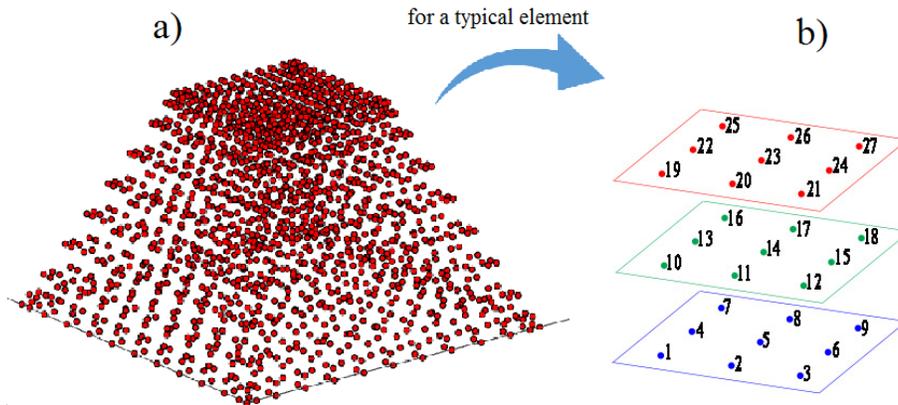


Figure 6: Control point volume array (a) and connectivity of control points in a typical quadratic element (b)

Listing 1: MATLAB code used to generate meshes in physical space

```

for e=1:numberNodes;
xi=nodeCoordinates(e,1);
eta=nodeCoordinates(e,2);
gamma=nodeCoordinates(e,3);

[shape,naturalDerivatives,Jacobianmatrix,elementDof]=shapeFunctionNURBS_3D_truncated(xi,eta,gamma,cpCoordinates,numberElementsX,numberElementsY,numberElementsZ,p,q,r,n,k,z);
spatter=elementDof(1:ndof);
counter=counter+1;
    geoX(counter)=shape'×cpCoordinates(spatter,1);
    geoY(counter)=shape'×cpCoordinates(spatter,2);
    geoZ(counter)=shape'×cpCoordinates(spatter,3);
end
geoCoordinates = [geoX(:) geoY(:) geoZ(:)];

```

3.2 Assembling stiffness and force matrices

The function *formStiffness_3D_Truncated_flexo* (Listing 2) returns the global stiffness matrix of the pyramid. The variable, *GDof* depicts the total number of degrees of freedom, which is four times the total number of control points. In fact, on each control point there are three components of the displacement field and an electric potential, $[u_x \ u_y \ u_z \ \theta]^T$. The solution field is also obtained as $[u_{x_1} \ u_{x_2} \ \dots \ u_{x_n} \ u_{y_1} \ u_{y_2} \ \dots \ u_{y_n} \ u_{z_1} \ u_{z_2} \ \dots \ u_{z_n} \ \theta_1 \ \theta_2 \ \dots \ \theta_n]^T$.

Listing 2: MATLAB code used to form the global stiffness matrix

```

Loop over elements
    Loop over Gauss points
stiffness(index_disp,index_disp)=stiffness(index_disp,index_disp)+(B_u'×C×B_u+Hessian_s×h_matrix×Hessian_s')×abs(wt×detJacob×J2);
stiffness(index_phi,index_phi)=stiffness(index_phi,index_phi) - B_electric×(Permittivity_matrix)×B_electric'×abs(wt×detJacob×J2);
stiffness(index_disp,index_phi)=stiffness(index_disp,index_phi)+(Hessian_u×flexo_matrix'×B_electric')×abs(wt×detJacob×J2);
stiffness(index_phi,index_disp)=stiffness(index_phi,index_disp)+(B_electric×flexo_matrix×Hessian_u')×abs(wt×detJacob×J2);
    end % loop over Gauss points
end %loop over elements

```

3.3 Derivatives of basis functions

As demonstrated in Fig. 1, there is a coordinate transformation in IGA in such a case that each point $\mathbf{x} = \{x, y, z\}$ in the physical space is mapped to a corresponding point $\boldsymbol{\xi} = \{\xi, \eta, \gamma\}$ in the parameter space and vice versa. In order to define the gradient (\mathbf{B}_u and \mathbf{B}_θ) and Hessian (\mathbf{H}_u and \mathbf{H}_s) matrices appearing in the weak form, we need to compute derivatives of the basis functions with respect to the physical coordinates. It takes the form

$$\begin{pmatrix} \frac{\partial^2 f}{\partial x^2} \\ \frac{\partial^2 f}{\partial y^2} \\ \frac{\partial^2 f}{\partial z^2} \\ \frac{\partial^2 f}{\partial y \partial z} \\ \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial x \partial y} \end{pmatrix} = [\mathbf{J}_{66}]^{-1} \begin{pmatrix} \frac{\partial^2 f}{\partial \xi^2} \\ \frac{\partial^2 f}{\partial \eta^2} \\ \frac{\partial^2 f}{\partial \gamma^2} \\ \frac{\partial^2 f}{\partial \eta \gamma} \\ \frac{\partial^2 f}{\partial \xi \gamma} \\ \frac{\partial^2 f}{\partial \xi \eta} \end{pmatrix} - [\mathbf{J}_{63}] \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix} \quad (16)$$

with $[\mathbf{J}_{66}]$ and $[\mathbf{J}_{63}]$ defined in **Appendix A**, where derivation of Eq. (16) is presented. The basis functions with their first and second natural (w.r.t ξ, η, γ) derivatives are returned by the `shapeFunctionNURBS_3D_truncated` function. Listing 3 converts natural into geometrical (w.r.t X, Y and Z coordinates) derivatives.

Listing 4 applies uniform pressure on the top of the pyramid. Firstly, the corresponding elements are picked up and then the pressure is applied by integrating the parameter `P_APPLIED_top` using 2-D Gauss quadrature method. The size of the force matrix is equal to the total number of degrees of freedom, `GDof`, and it is assembled as $\mathbf{Force} = [f_{x_1} \ f_{x_2} \ \dots \ f_{x_n} \ f_{y_1} \ f_{y_2} \ \dots \ f_{y_n} \ f_{z_1} \ f_{z_2} \ \dots \ f_{z_n} \ f_{\theta_1} \ f_{\theta_2} \ \dots \ f_{\theta_n}]^T$.

Listing 3: MATLAB code used to convert natural into geometrical derivatives

```

invJacobian=inv(Jacobianmatrix);
detJacob=det(Jacobianmatrix);
XYderivatives=naturalDerivatives×invJacobian;
J2 = ((nodeCoordinatesX(e1+1)-nodeCoordinatesX(e1)) × (nodeCoordinatesY(e2+1)-
nodeCoordinatesY(e2)) × (nodeCoordinatesZ(e3+1)-nodeCoordinatesZ(e3)) )/8;

dervX_xi=Jacobianmatrix(1,1);      dervY_xi=Jacobianmatrix(2,1);
dervX_eta=Jacobianmatrix(1,2);    dervY_eta=Jacobianmatrix(2,2);
dervX_chi=Jacobianmatrix(1,3);    dervY_chi=Jacobianmatrix(2,3);

dervZ_xi=Jacobianmatrix(3,1);     dervZ_eta=Jacobianmatrix(3,2);
dervZ_chi=Jacobianmatrix(3,3);

J_6_6=[dervX_xi^2;   dervX_eta^2;   dervX_chi^2;   dervX_eta×dervX_chi;
dervX_xi×dervX_chi; dervX_xi×dervX_eta; dervY_xi^2; dervY_eta^2; dervY_chi^2;
dervY_eta×dervY_chi; dervY_xi×dervY_chi; dervY_xi×dervY_eta; dervZ_xi^2;
dervZ_eta^2;   dervZ_chi^2;   dervZ_eta×dervZ_chi;   dervZ_xi×dervZ_chi;
dervZ_xi×dervZ_eta; 2×dervZ_xi×dervY_xi;
2×dervZ_eta×dervY_eta; 2×dervZ_chi×dervY_chi; (dervY_eta×dervZ_chi+dervY_chi
×dervZ_eta);
(dervY_xi×dervZ_chi+dervY_chi×dervZ_xi); (dervY_xi×dervZ_eta+dervY_eta×derv
Z_xi);
2×dervZ_xi×dervX_xi;   2×   dervZ_eta×dervX_eta;   2×dervZ_chi×dervX_chi;
(dervX_eta×dervZ_chi+dervX_chi×dervZ_eta); (dervX_xi×dervZ_chi+dervX_chi×derv
Z_xi);   (dervX_xi×dervZ_eta+dervX_eta×dervZ_xi);   2×dervX_xi×dervY_xi;
2×dervX_eta×dervY_eta;   2×dervX_chi×dervY_chi;
(dervX_eta×dervY_chi+dervX_chi×dervY_eta);
(dervX_xi×dervY_chi+dervX_chi×dervY_xi); (dervX_xi×dervY_eta+dervX_eta×derv
Y_xi)];

XYderivatives_second=(naturalDerivatives_second-
XYderivatives×Jacobianmatrix_3_6)×inv(J_6_6);

```

Listing 4: MATLAB code used to apply uniform pressure on the top of the pyramid

```

[W,Q]=quadrature(noGPS_REDUCED, 'GAUSS', 2); % noGPs×noGPs point
quadrature
Loop over top elements
for gp=1:size(W,1);
    pt=Q(gp,:);    wt=W(gp);    ptxi=pt(1);    pteta=pt(2);

    xi=(nodeCoordinatesX(1,e1))+0.5×(ptxi+1)×(nodeCoordinatesX(1,e1+1)-
nodeCoordinatesX(1,e1));
    eta=(nodeCoordinatesY(1,e2))+0.5×(pteta+1)×(nodeCoordinatesY(1,e2+1)-
nodeCoordinatesY(1,e2));

    % 2D shape functions and derivatives
    [shape,naturalDerivatives,naturalDerivatives_second,Jacobianmatrix,Jacobianmatrix_
2_3,elementDof]=shapeFunctionNURBS_2D(xi,eta,cpCoordinates(topNodes,1:2),nu
mberElementsX,numberElementsY,p,q,nnel,n,k);

    % Jacobian matrix, inverse of Jacobian, derivatives w.r.t. x,y
    invJacobian=inv(Jacobianmatrix);
    detJacob=det(Jacobianmatrix);
    XYderivatives=naturalDerivatives×invJacobian;
    J2 = ((nodeCoordinatesX(e1+1)-nodeCoordinatesX(e1))×(nodeCoordinatesY(e2+1)-
nodeCoordinatesY(e2)))/4;

    indice_pressure_force=elementDof(1:ndof_xy)    +z×(numberControlpoints_xy)+
2×numberControlpoints ; % on top surface in Z direction

    % Force vector
    force(indice_pressure_force)=force(indice_pressure_force)+shape×P_APPLIED_top×
wt×detJacob×J2;
    end
end %loop over top elements

```

3.4 Boundary conditions

To apply the equipotential boundary condition on the top surface of the pyramid, penalty stiffness method is used as described in Listing 5. The electric potential degrees of freedom corresponding to the top nodes are tied to each other one by one in both X and Y directions. The script *applyBC_3D* (see Listing 6) is used to impose mechanical boundary conditions as well as the zero-electric potential boundary condition on the bottom edge.

Listing 5: MATLAB code used to apply equipotential electrical boundary condition

```

DUMMY_AA=reshape((topNodes),n+1,k+1);
DUMMY_BB=reshape((topNodes),n+1,k+1)';
for j=1:k+1;
for i=1:n;
sctr_x = [DUMMY_BB(j,i) DUMMY_BB(j,i+1)]+3×numberControlpoints;
sctr_y = [DUMMY_AA(j,i) DUMMY_AA(j,i+1)]+3×numberControlpoints;
w_phi =100000;
penaltyStiffness_phi = w_phi×[1 -1;-1 1];
stiffness(sctr_x,sctr_x) = stiffness(sctr_x,sctr_x) + penaltyStiffness_phi;
stiffness(sctr_y,sctr_y) = stiffness(sctr_y,sctr_y) + penaltyStiffness_phi;
end
end

```

Listing 6: MATLAB code used to apply ground electrical and mechanical BC

```

bcwt=mean(diag(stiffness)); % a measure of the average size of an element in K
% used to keep the conditioning of the K matrix
force=force-stiffness(:,udofs)×uFixed; % modify the force vector
force=force-stiffness(:,vdofs)×vFixed;      force=force-stiffness(:,wdofs)×wFixed;
force=force-stiffness(:,phidofs)×phiFixed;  force(udofs) = bcwt×uFixed;
force(vdofs) = bcwt×vFixed;                  force(wdofs) = bcwt×wFixed;
force(phidofs) = bcwt×phiFixed;
stiffness(udofs,:)=0; % zero out the rows and columns of the K matrix
stiffness(vdofs,:)=0; stiffness(wdofs,:)=0; stiffness(phidofs,:)=0;
stiffness(:,udofs)=0; stiffness(:,vdofs)=0; stiffness(:,wdofs)=0;
stiffness(:,phidofs)=0;
stiffness(udofs,udofs)=bcwt×speye(length(udofs)); % put ones×bcwt on the
diagonal
stiffness(vdofs,vdofs)=bcwt×speye(length(vdofs));
stiffness(wdofs,wdofs)=bcwt×speye(length(wdofs));
stiffness(phidofs,phidofs)=bcwt×speye(length(phidofs));

```

3.5 Post processing

The function *VTKPostProcess3d*, is adopted to visualize the numerical results using *ParaView* which is an open-source, multi-platform data analysis and visualization platform.

4 Results and discussions

Fig. 7 shows the simulation results for the inclination angle, $\alpha = \frac{\pi}{4}$ with different area ratios, $R = \left(\frac{a_2}{a_1}\right)^2$. The pressure is set so that to apply the load of $F = 200\text{ N}$ for all values of R . The electric potential and strain through thickness direction are plotted for each case. The results show acceptable conformity with results presented in Abdollahi et al. [Abdollahi, Millan, Peco et al. (2015)]. It is observable that an increase in R causes larger strain gradients and consequently more electric polarization. The same trend is also found for $\alpha = \frac{\pi}{3}$ as shown in Fig. 8.

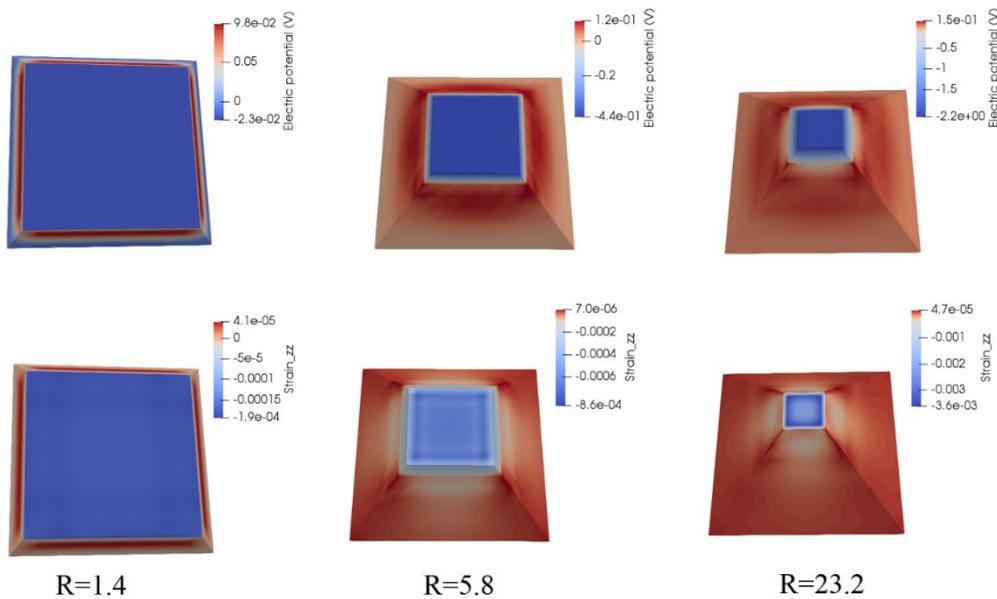


Figure 7: Distribution of the electric potential and strain in thickness direction for different values of area ratio, $R = \left(\frac{a_2}{a_1}\right)^2$. In all insets $\alpha = \frac{\pi}{4}$ and $F = 200\text{ N}$

4.1 Extensions

A suite of extensions can be thought for the presented code. For instance, change in supporting and boundary conditions can be mentioned. Flexible supports instead of rigid type can cause bending of the pyramid, besides its compression. The inverse flexoelectric effect can be investigated by applying electric potential on the electrodes and measure the deformation of the pyramid. A flexoelectric multi-pyramid composite can be designed using multi patches.

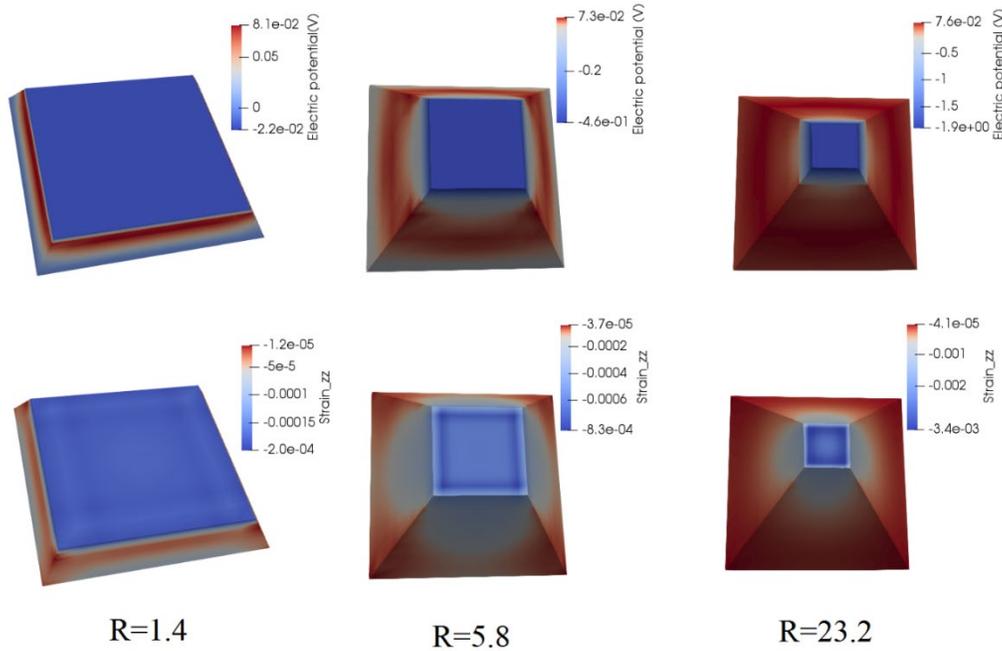


Figure 8: Distribution of the electric potential and strain in thickness direction for different values of area ratio, $R = \left(\frac{a_2}{a_1}\right)^2$. In all insets $\alpha = \frac{\pi}{3}$ and $F = 200\text{ N}$

5 Concluding remarks

We present a three-dimensional isogeometric formulations of flexoelectricity with its MATLAB implementation for a truncated pyramid setup. We take advantages of the B-spline basis functions to discretize the fourth order partial differential equations of flexoelectricity; which demand at least C^1 continuous basis functions in a Galerkin method. Our numerical simulation clarifies that the strain gradients are highly localized around the edges of the pyramid and their magnitudes increase by an increase in the pyramid area ratio, R . We provide MATLAB code as Supplementary Materials of the paper with explanations to facilitate the popularity of the topic.

Acknowledgment: Hamid Ghasemi acknowledge the support of the Mechanical Engineering department at Arak University of Technology. Xiaoying Zhuang gratefully acknowledge the financial support by European Research Council for COTOFLEXI project (802205). Harold Park acknowledges the support of the Mechanical Engineering department at Boston University. Timon Rabczuk gratefully acknowledge financial support by the 2019 Foreign Experts Plan of Hebei Province.

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

References

- Abdollahi, A.; Millan, D.; Peco, C.; Arroyo, M.; Arias, I.** (2015): Revisiting pyramid compression to quantify flexoelectricity: a three-dimensional simulation study. *Physical Review B*, vol. 91 pp. 104103.
- Abdollahi, A.; Peco, C.; Millan, D.; Arroyo, M.; Arias, I.** (2014): Computational evaluation of the flexoelectric effect in dielectric solids. *Journal of Applied Physics*, vol. 116, pp. 093502.
- Abdollahi, A.; Peco, C.; Millan, D.; Arroyo, M.; Catalan, G. et al.** (2015): Fracture toughening and toughness asymmetry induced by flexoelectricity. *Physical Review B*, vol. 92, no. 9, pp. 094101.
- Codony, D.; Marco, O.; Fernandez-Mendez, S.; Arias, I.** (2019) An immersed boundary hierarchical B-spline method for flexoelectricity. *Journal of the Mechanics and Physics of Solids*, vol. 354, pp. 750-782.
- Cross, L. E.** (2006): Flexoelectric effects: charge separation in insulating solids subjected to elastic strain gradients. *Journal of Materials Science*, vol. 41, pp. 53-63.
- Deng, F.; Deng, Q.; Shen, S.** (2018): A 3D mixed finite element for flexoelectricity. *Journal of Applied Mechanics*, vol. 85, pp. 031009-1-10.
- Eliseev, E. A.; Morozovska, A. N.; Glinchuk, M. D.; Blinc, R.** (2009): Spontaneous flexoelectric/flexomagnetic effect in nanoferroics. *Physical Review B*, vol. 79, pp. 165433.
- Ghasemi, H.; Park, H. S.; Rabczuk, T.** (2017): A level-set based IGA formulation for topology optimization of flexoelectric materials. *Computer Methods in Applied Mechanics and Engineering*, vol. 313, pp. 239-258.
- Ghasemi, H.; Park, H. S.; Rabczuk, T.** (2018): A multi-material level set-based topology optimization of flexoelectric composites. *Computer Methods in Applied Mechanics and Engineering*, vol. 332, pp. 47-62.
- He, B.; Javvaji, B.; Zhuang, X.** (2019): Characterizing flexoelectricity in composite material using the element-free Galerkin method. *Energies*, vol. 12, no. 2, pp. 271-289.
- Huang, W.; Shu, L.; Kwon, S. R.; Zhang, S.; Yuan, F. G.; Jiang, X.** (2014): Fabrication and measurement of a flexoelectric micro-pyramid composite. *AIP Advances*, vol. 4, pp. 127115.
- Hughes, T. J. R.; Cottrell, J. A.; Bazilevs, Y.** (2005): Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, vol. 194, pp. 4135-4195.
- Kwon, S. R.** (2017): Structural analysis of truncated pyramids for flexoelectric sensing. *Journal of Mechanical Science and Technology*, vol. 31, no. 12, pp. 5971-5975.
- Liu, C.; Wang, J.; Xu, G.; Kamlah, M.; Zhang, T. Y.** (2018): An isogeometric approach to flexoelectric effect in ferroelectric materials. *International Journal of Solids and Structures*, vol. 162, pp. 198-210.
- Mao, S.; Purohit, P. K.** (2014): Insights into flexoelectric solids from strain-gradient elasticity. *Journal of Applied Mechanics*, vol. 81, pp. 081004.
- Mao, S.; Purohit, P. K.; Aravas, N.** (2016): Mixed finite-element formulations in

piezoelectricity and flexoelectricity. *Proceedings of the Royal Society A*, vol. 472, pp. 20150879.

Mao, Y.; Ai, S.; Xiang, X.; Chen, C. (2016): Theory for dielectrics considering the direct and converse flexoelectric effects and its finite element implementation. *Applied Mathematical Modelling*, vol. 40, no. 15, pp. 7115-7137.

Maranganti, R.; Sharma, N. D.; Sharma, P. (2006): Electromechanical coupling in nonpiezoelectric materials due to nanoscale nonlocal size effects: Green's function solutions and embedded inclusions. *Physical Review B*, vol. 74, pp. 014110.

Nguyen, B. H.; Zhuang, X.; Rabczuk, T. (2018): Numerical model for the characterization of Maxwell-Wagner relaxation in piezoelectric and flexoelectric composite material. *Computers and Structures*, vol. 208, pp. 75-91.

Poya, R.; Gil, A. J.; Ortigosa, R.; Palma, R. (2019): On a family of numerical models for couple stress based flexoelectricity for continua and beams. *Journal of the Mechanics and Physics of Solids*, vol. 125, pp. 613-652.

Qi, L.; Huang, S.; Fu, G.; Zhou, S.; Jiang, X. (2018): On the mechanics of curved flexoelectric microbeams. *International Journal of Engineering Science*, vol. 124, pp. 1-15.

Sharma, N. D.; Landis, C. M.; Sharma, P. (2010): Piezoelectric thin-film superlattices without using piezoelectric materials. *Journal of Applied Physics*, vol. 108, pp. 024304.

Sharma, N. D.; Maranganti, R.; Sharma, P. (2007): On the possibility of piezoelectric nanocomposites without using piezoelectric materials. *Journal of the Mechanics and Physics of Solids*, vol. 55, pp. 2328-2350.

Shen, S.; Hu, S. (2010): A theory of flexoelectricity with surface effect for elastic dielectrics. *Journal of the Mechanics and Physics of Solids*, vol. 58, pp. 665-677.

Sidhardh, S.; Ray, M. C. (2018): Effective properties of flexoelectric fiber-reinforced nanocomposite. *Materials Today Communications*, vol. 17, pp.114-123.

Thai, T. Q.; Rabczuk, T.; Zhuang, X. (2018): A large deformation isogeometric approach for flexoelectricity and soft materials. *Computer Methods in Applied Mechanics and Engineering*, vol. 341, pp. 718-739.

Yudin, P. V.; Tagantsev, A. K. (2013): Topical review: fundamentals of flexoelectricity in solids. *Nanotechnology*, vol. 24, pp. 432001-432037.

Yvonnet, J.; Liu, L. (2017): A numerical framework for modeling flexoelectricity and Maxwell stress in soft dielectrics at finite strains. *Computer Methods in Applied Mechanics and Engineering*, vol. 313, pp. 450-482.

Zhu, W. Y.; Fu, J. Y.; Li, N.; Cross, L. (2006): Piezoelectric composite based on the enhanced flexoelectric effects. *Applied Physics Letters*, vol. 89, pp. 192904.

Appendix A

Using the chain rules of partial differentiation for a given function $f(\mathbf{x}(\boldsymbol{\xi}))$ we can write:

$$\begin{cases} \frac{\partial f}{\partial \xi} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial \xi} + \frac{\partial f}{\partial z} \frac{\partial z}{\partial \xi} \\ \frac{\partial f}{\partial \eta} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial \eta} + \frac{\partial f}{\partial z} \frac{\partial z}{\partial \eta} \\ \frac{\partial f}{\partial \gamma} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial \gamma} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial \gamma} + \frac{\partial f}{\partial z} \frac{\partial z}{\partial \gamma} \end{cases} \quad (\text{A1})$$

in the matrix form it takes the form:

$$\begin{pmatrix} \frac{\partial f}{\partial \xi} \\ \frac{\partial f}{\partial \eta} \\ \frac{\partial f}{\partial \gamma} \end{pmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \gamma} & \frac{\partial y}{\partial \gamma} & \frac{\partial z}{\partial \gamma} \end{bmatrix} \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix} \quad (\text{A2})$$

The 3×3 matrix is called \mathbf{J} which is the Jacobian matrix of the transformation. For a quadratic B-spline basis functions corresponding to an element with 9 control points, we have 9 basis functions f_1, f_2, \dots, f_9 . Thus,

$$\begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_2}{\partial x} & \dots & \frac{\partial f_9}{\partial x} \\ \frac{\partial f_1}{\partial y} & \frac{\partial f_2}{\partial y} & \dots & \frac{\partial f_9}{\partial y} \\ \frac{\partial f_1}{\partial z} & \frac{\partial f_2}{\partial z} & \dots & \frac{\partial f_9}{\partial z} \end{pmatrix} = \mathbf{J}^{-1} \begin{pmatrix} \frac{\partial f_1}{\partial \xi} & \frac{\partial f_2}{\partial \xi} & \dots & \frac{\partial f_9}{\partial \xi} \\ \frac{\partial f_1}{\partial \eta} & \frac{\partial f_2}{\partial \eta} & \dots & \frac{\partial f_9}{\partial \eta} \\ \frac{\partial f_1}{\partial \gamma} & \frac{\partial f_2}{\partial \gamma} & \dots & \frac{\partial f_9}{\partial \gamma} \end{pmatrix} \quad (\text{A3})$$

To calculate the second derivatives we write:

$$\frac{\partial^2 f}{\partial \xi^2} = \frac{\partial}{\partial \xi} \left(\frac{\partial f}{\partial \xi} \right) = \frac{\partial}{\partial \xi} \left(\frac{\partial f}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial \xi} + \frac{\partial f}{\partial z} \frac{\partial z}{\partial \xi} \right) = \frac{\partial^2 f}{\partial x \partial \xi^2} + \frac{\partial^2 f}{\partial \xi \partial x \partial \xi} + \frac{\partial^2 f}{\partial y \partial \xi^2} + \frac{\partial^2 f}{\partial \xi \partial y \partial \xi} + \frac{\partial^2 f}{\partial z \partial \xi^2} + \frac{\partial^2 f}{\partial \xi \partial z \partial \xi} \quad (\text{A4})$$

by extending $\frac{\partial^2 f}{\partial \xi \partial x}$ we have

$$\frac{\partial^2 f}{\partial \xi \partial x} = \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial \xi} \right) = \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial \xi} + \frac{\partial f}{\partial z} \frac{\partial z}{\partial \xi} \right) = \frac{\partial^2 f}{\partial x^2 \partial \xi} + 0 + \frac{\partial^2 f}{\partial x \partial y \partial \xi} + 0 + \frac{\partial^2 f}{\partial x \partial z \partial \xi} + 0 = \frac{\partial^2 f}{\partial x^2 \partial \xi} + \frac{\partial^2 f}{\partial x \partial y \partial \xi} + \frac{\partial^2 f}{\partial x \partial z \partial \xi} \quad (\text{A4a})$$

analogously we can write

$$\frac{\partial^2 f}{\partial \xi \partial y} = \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial \xi} \right) = \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial \xi} + \frac{\partial f}{\partial z} \frac{\partial z}{\partial \xi} \right) = \frac{\partial^2 f}{\partial y \partial x \partial \xi} + 0 + \frac{\partial^2 f}{\partial y^2 \partial \xi} + 0 + \frac{\partial^2 f}{\partial y \partial z \partial \xi} + 0 = \frac{\partial^2 f}{\partial y^2 \partial \xi} + \frac{\partial^2 f}{\partial y \partial x \partial \xi} + \frac{\partial^2 f}{\partial y \partial z \partial \xi} \quad (\text{A4b})$$

$$\frac{\partial^2 f}{\partial \xi \partial z} = \frac{\partial}{\partial z} \left(\frac{\partial f}{\partial \xi} \right) = \frac{\partial}{\partial z} \left(\frac{\partial f}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial \xi} + \frac{\partial f}{\partial z} \frac{\partial z}{\partial \xi} \right) = \frac{\partial^2 f}{\partial z \partial x \partial \xi} + 0 + \frac{\partial^2 f}{\partial z \partial y \partial \xi} + 0 + \frac{\partial^2 f}{\partial z^2 \partial \xi} + 0 = \frac{\partial^2 f}{\partial z^2 \partial \xi} + \frac{\partial^2 f}{\partial z \partial x \partial \xi} + \frac{\partial^2 f}{\partial z \partial y \partial \xi} \quad (\text{A4c})$$

$$\frac{\partial^2 f}{\partial \eta \partial x} = \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial \eta} \right) = \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial \eta} + \frac{\partial f}{\partial z} \frac{\partial z}{\partial \eta} \right) = \frac{\partial^2 f}{\partial x^2 \partial \eta} + 0 + \frac{\partial^2 f}{\partial x \partial y \partial \eta} + 0 + \frac{\partial^2 f}{\partial x \partial z \partial \eta} + 0 = \frac{\partial^2 f}{\partial x^2 \partial \eta} + \frac{\partial^2 f}{\partial x \partial y \partial \eta} + \frac{\partial^2 f}{\partial x \partial z \partial \eta} \quad (\text{A4d})$$

$$\begin{aligned}
 & \left[\begin{array}{c} \frac{\partial^2 f}{\partial \xi^2} \\ \frac{\partial^2 f}{\partial \eta^2} \\ \frac{\partial^2 f}{\partial \gamma^2} \\ \frac{\partial^2 f}{\partial \eta \gamma} \\ \frac{\partial^2 f}{\partial \xi \gamma} \\ \frac{\partial^2 f}{\partial \xi \eta} \end{array} \right] = \\
 & \left[\begin{array}{cccccc} \left(\frac{\partial x}{\partial \xi}\right)^2 & \left(\frac{\partial y}{\partial \xi}\right)^2 & \left(\frac{\partial z}{\partial \xi}\right)^2 & 2 \frac{\partial z}{\partial \xi} \frac{\partial y}{\partial \xi} & 2 \frac{\partial z}{\partial \xi} \frac{\partial x}{\partial \xi} & 2 \frac{\partial y}{\partial \xi} \frac{\partial x}{\partial \xi} \\ \left(\frac{\partial x}{\partial \eta}\right)^2 & \left(\frac{\partial y}{\partial \eta}\right)^2 & \left(\frac{\partial z}{\partial \eta}\right)^2 & 2 \frac{\partial z}{\partial \eta} \frac{\partial y}{\partial \eta} & 2 \frac{\partial z}{\partial \eta} \frac{\partial x}{\partial \eta} & 2 \frac{\partial y}{\partial \eta} \frac{\partial x}{\partial \eta} \\ \left(\frac{\partial x}{\partial \gamma}\right)^2 & \left(\frac{\partial y}{\partial \gamma}\right)^2 & \left(\frac{\partial z}{\partial \gamma}\right)^2 & 2 \frac{\partial z}{\partial \gamma} \frac{\partial y}{\partial \gamma} & 2 \frac{\partial z}{\partial \gamma} \frac{\partial x}{\partial \gamma} & 2 \frac{\partial y}{\partial \gamma} \frac{\partial x}{\partial \gamma} \\ \frac{\partial x}{\partial \eta} \frac{\partial x}{\partial \gamma} & \frac{\partial y}{\partial \eta} \frac{\partial y}{\partial \gamma} & \frac{\partial z}{\partial \eta} \frac{\partial z}{\partial \gamma} & \frac{\partial z}{\partial \eta} \frac{\partial y}{\partial \gamma} + \frac{\partial y}{\partial \eta} \frac{\partial z}{\partial \gamma} & \frac{\partial z}{\partial \eta} \frac{\partial x}{\partial \gamma} + \frac{\partial x}{\partial \eta} \frac{\partial z}{\partial \gamma} & \frac{\partial y}{\partial \eta} \frac{\partial x}{\partial \gamma} + \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \gamma} \\ \frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \gamma} & \frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \gamma} & \frac{\partial z}{\partial \xi} \frac{\partial z}{\partial \gamma} & \frac{\partial z}{\partial \xi} \frac{\partial y}{\partial \gamma} + \frac{\partial y}{\partial \xi} \frac{\partial z}{\partial \gamma} & \frac{\partial z}{\partial \xi} \frac{\partial x}{\partial \gamma} + \frac{\partial x}{\partial \xi} \frac{\partial z}{\partial \gamma} & \frac{\partial y}{\partial \xi} \frac{\partial x}{\partial \gamma} + \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \gamma} \\ \frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \xi} \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \xi} \frac{\partial y}{\partial \eta} + \frac{\partial y}{\partial \xi} \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \xi} \frac{\partial x}{\partial \eta} + \frac{\partial x}{\partial \xi} \frac{\partial z}{\partial \eta} & \frac{\partial y}{\partial \xi} \frac{\partial x}{\partial \eta} + \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} \end{array} \right] \left[\begin{array}{c} \frac{\partial^2 f}{\partial x^2} \\ \frac{\partial^2 f}{\partial y^2} \\ \frac{\partial^2 f}{\partial z^2} \\ \frac{\partial^2 f}{\partial y \partial z} \\ \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial x \partial y} \end{array} \right] + \\
 & \left[\begin{array}{ccc} \frac{\partial^2 x}{\partial \xi^2} & \frac{\partial^2 y}{\partial \xi^2} & \frac{\partial^2 z}{\partial \xi^2} \\ \frac{\partial^2 x}{\partial \eta^2} & \frac{\partial^2 y}{\partial \eta^2} & \frac{\partial^2 z}{\partial \eta^2} \\ \frac{\partial^2 x}{\partial \gamma^2} & \frac{\partial^2 y}{\partial \gamma^2} & \frac{\partial^2 z}{\partial \gamma^2} \\ \frac{\partial^2 x}{\partial \eta \partial \gamma} & \frac{\partial^2 y}{\partial \eta \partial \gamma} & \frac{\partial^2 z}{\partial \eta \partial \gamma} \\ \frac{\partial^2 x}{\partial \xi \partial \gamma} & \frac{\partial^2 y}{\partial \xi \partial \gamma} & \frac{\partial^2 z}{\partial \xi \partial \gamma} \\ \frac{\partial^2 x}{\partial \xi \partial \eta} & \frac{\partial^2 y}{\partial \xi \partial \eta} & \frac{\partial^2 z}{\partial \xi \partial \eta} \end{array} \right] \left\{ \begin{array}{c} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{array} \right\} \tag{A6}
 \end{aligned}$$

Finally, one can compute the derivatives with respect to physical coordinates as:

$$\begin{pmatrix} \frac{\partial^2 f}{\partial x^2} \\ \frac{\partial^2 f}{\partial y^2} \\ \frac{\partial^2 f}{\partial z^2} \\ \frac{\partial^2 f}{\partial y \partial z} \\ \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial x \partial y} \end{pmatrix} = [\mathbf{J}_{66}]^{-1} \begin{pmatrix} \frac{\partial^2 f}{\partial \xi^2} \\ \frac{\partial^2 f}{\partial \eta^2} \\ \frac{\partial^2 f}{\partial \gamma^2} \\ \frac{\partial^2 f}{\partial \eta \gamma} \\ \frac{\partial^2 f}{\partial \xi \gamma} \\ \frac{\partial^2 f}{\partial \xi \eta} \end{pmatrix} - [\mathbf{J}_{63}] \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix} \quad (\text{A7})$$