Noname manuscript No.
(will be inserted by the editor)

MATLAB Codes for 3D Topology Optimization of Multi-Material Piezoelectric Actuators and Energy Harvesters

Abbas Homayouni-Amlashi [1] · Ole Sigmund [2] · Thomas Schlinquer [1,3]
Micky Rakotondrabe [4] · Abdenbi Mohand-Ousaid [1]

Received: date / Accepted: date

Abstract This paper presents two MATLAB codes for topology optimization of multi-material piezoelectric actuators and energy harvesters. These codes provide the extensions of the previously published 2D topology optimization codes for piezoelectric actuators and energy harvesters (Struct Multidisc Optim 63 (2), 983-1014) with two major contributions: 1) extension to the third dimension, 2) combination of piezoelectric (active) and non-piezoelectric (passive) materials in the design domain. The codes are written in the most flexible form to be compatible with different optimization problems and practical case studies of piezoelectricity that exist in the literature. The codes address unique challenges that emerge by introducing the third dimension to non-isotropic piezoelectric materials including the polarization direction and definition of electrodes. The finite element discretization has been done with two different types of 3D hexahedral elements: 1) 8 node trilinear elements, 2) 20 node quadratic elements. The users are free to choose between these element types for the finite element model of the structure based on having preferences for accuracy or computation time. A new method of indexing the elements, nodes and degrees of freedom is introduced to facilitate the definition of loads, boundary conditions, electrodes, etc. The inclusion of piezoelectric material and non-piezoelectric material in the design domain is by default. In comparison to previously published 2D codes, the codes in this paper benefit from the latest advancements in optimization algorithms, filtering methods and speedup techniques. The codes are independent and hence can be run without calling any external code. Different parts of the codes are explained in detail to make them comprehensive for newcomers in the field of topology optimization of piezoelectric structures.

Keywords Topology optimization · MATLAB code · Piezoelectric actuator · Piezoelectric energy harvester · Finite element model

1 Introduction

Topology Optimization (TO) provides the optimized material distribution inside a design domain when there is no prior knowledge of the final optimal layout of the structure. TO can obtain optimized designs which are impossible to obtain by intuitive or trial-error methods.

Acknowledgement of fundings: This work has been supported by 1- The national CODE-TRACK project (ANR-17-CE05-0014-01, Control theory tools for optimal design of piezoelectric energy harvesters devoted to birds tracking devices), 2- OptoBots project (ANR-21-CE33-0003), 3- EIPHI Graduate School (contract ANR-17-EURE-0002) 4- Villum Investigator project AMSTRAD (VIL54487) from VILLUM FONDEN

A. Homayouni-Amlashi
E-mail: abbas.homayouni@gmail.com

O. Sigmund
E-mail: olsi@dtu.dk

T. Schlinquer
E-mail: thomas.schlinquer@eseo.fr

M. Rakotondrabe
E-mail: mrakoton@enit.fr

A. Mohand-Ousaid
E-mail: abdenbi.mohand@femto-st.fr

[1] FEMTO-ST Institute, Université Bourgogne Franche-Comté, CNRS, Besançon 25000, France.
[2] Department of Civil and Mechanical Engineering, Technical University of Denmark, Koppels Alle 404, 2800 Kongens Lyngby, Denmark
[3] ESEO engineering school, Dijon, France
[4] Laboratoire Génie de Production, National School of Engineering in Tarbes (ENIT), Toulouse INP, University of Toulouse, 47, Avenue d'Azereix, Tarbes, France

As such, this algorithmic design methodology has revolutionized conventional structural design approaches. The current understanding of TO has been introduced to the society of structural optimization by Bendsøe and Kikuchi (1988) in which the homogenization approach was used. Later, several other approaches have been proposed for implementation of TO, including the Solid Isotropic Material with Penalization (SIMP) (Bendsøe, 1989; Rozvany et al., 1992), Evolutionary structural optimization (ESO) (Xie and Steven, 1993; Xia et al., 2018), level set as reviewed in (van Dijk et al., 2013; Andreasen et al., 2020) and Moving Morphable Components (MMC) (Guo et al., 2014; Zhang et al., 2017). A detailed review of these approaches has been done by Sigmund and Maute (2013) and Deaton and Grandhi (2014). Among the aforementioned approaches, the SIMP method is the most prominent and well-developed approach due to its efficiency and ease of implementation. SIMP is a density-based approach in which the design domain is discretized by a finite number of elements and continuous optimization variables are attributed to each of these elements. The optimization variables are steered to their bounds through iterations using the updating algorithms and penalization of intermediate densities. In this manner, the design domain evolves to the optimized topology based on the desired objective function and the given constraints. The studies on TO with different approaches initially have been done on the compliance problem of mechanical structures (Bendsøe and Sigmund, 2003) or compliant mechanisms (Sigmund, 1997). Due to the success of the TO in these fields, it was later developed to other physics including fluid dynamics (Alexandersen and Andreasen, 2020), heat transfer (Dbouk, 2017), photonics (Christiansen and Sigmund, 2021), electromagnetics (Li et al., 2020), electrostatics (Homayouni-Amlashi et al., 2024a) and piezoelectricity (Homayouni-Amlashi et al., 2019, 2024b, 2023b,a; Yang et al., 2022).

Piezoelectric materials are very attractive in industry and research environments due to their high bandwidth, high resolution and high force density. Piezoelectric materials have wide applications in industrial actuators and sensors, microphones, nanopositioners in atomic force microscopy, robotics, energy harvesting, etc (Sekhar et al., 2021). There is no review paper on the application of TO to piezoelectricity. Therefore, we will do a brief review here to outline the current status of the ongoing research. Primarily, the homogenization approach is used for piezoelectric composites exhibiting periodic structure where effective properties are evaluated by homogenization (Gałka et al., 1992). Silva et al. (1997) used the same approach to design optimal piezoelectric microstructures and transducers (Silva, 1998; Silva et al., 1999; Silva and Kikuchi, 1999). Du et al. (2000) considered dynamic actuation with piezoelectric stacks while using the variable element density approach to maximize the stroke. Sigmund et al. (1998) used SIMP approach to design the cell structure of a piezo-composite to increase the 31-mode efficiency. Carbonari et al. (2005) considered the piezoelectric and passive materials inside the design domain simultaneously to design actuators that produce desired displacement at different locations with different directions. Kögl and Silva (2005) proposed an extension of the SIMP method called PEMAP-P (piezoelectric material with penalization and polarization) in which the polarization direction of the piezoelectric material has been optimized in addition to its layout. Similar work has been done by Donoso and Bellido (2009) for topology optimization of piezoelectric modal sensors/actuators by optimization of polarization profile. Kang and Wang (2010) did the topology optimization on the multi-layer piezoelectric bending actuators where the distribution of voltage is optimized simultaneously with the material distribution. Rupp et al. (2009, 2008) for the first time applied TO for piezoelectric energy harvesters (PEH) while electrical circuit coupling is also considered in the modeling. Zheng et al. (2009) performed static topology optimization for PEH by performing detailed sensitivity analysis and utilization of 3D FEM to model the system. Noh and Yoon (2012) extended the work of Zheng et al. (2009) to TO of PEH under dynamic load and the effect of penalization factors and frequency of excitation on the final results are investigated case by case. Lin et al. (2011) optimized the PEH under the application of a broadband random vibration while optimizing the place of a lumped mass. Wein et al. (2013) considered constraint on von Mises stress in the problem formulation. Donoso and Sigmund (2016) considered the null polarity in the optimization of polarization direction where the width of the null polarization is enforced. Salas et al. (2018) utilized the PEMAP-P to optimize for the transient response in addition to harmonic response. Although the consideration of polarization in the optimization highly improved the performance of the PEH by preventing the charge cancellation problem, the fabrication and realization of the piezoelectric device are complicated. The reason is due to emergence of islands in the optimized polarization profiles. Therefore, Donoso and Guest (2019) proposed an interesting method to address the connectivity issue in the optimization of electrodes and to avoid the island problem which simplifies the fabrication process. Homayouni-Amlashi et al. (2020) extended the works by Zheng et al. (2009), Noh and Yoon (2012) to design multidirectional single layer PEH that can harvest the energy from 2D planar forces coming from different directions. Homayouni-Amlashi et al. (2019) continued the work by considering two-layer piezoelectric plate to design a multi directional PEH that can harvest the energy coming from different directions in 3D space. In these researches, the polarization profile of the PEHs is optimized along their topology using the PEMAP-P approach.

Knowing the fact that the best efficiency of a PEH can be obtained when it is excited at its resonance frequency, researchers considered the optimization of eigenfrequencies and eigenmodes of the PEHs. Kim and Shin (2013) did eigenfrequency optimization to increase the electromechanical coupling coefficient of the design and to match the resonance frequency and the excitation frequency. Nakasone and Silva (2010) formulated the optimization problem to optimize the eigenmodes in addition to optimization of the eigenfrequencies of the piezoelectric resonator. Wein et al. (2009) also did dynamic TO for a piezoelectric actuator with multiple frequency optimization. Wang et al. (2017) optimized the shape of the eigenmodes by optimization of the electromechanical coupling coefficient through modal responses. Recently, Homayouni-Amlashi et al. (2023a) considered frequency tuning in TO for the multi directional PEH by considering the mass of attachment as an optimization variable.

Different forms of multi-material TO are also applied to piezoelectric devices. In the work by Sigmund (2001b), Carbonari et al. (2005) and Molter et al. (2016) three phases of material are considered inside the design domain based on the defined material interpolation function: 1- Active (piezo), 2- Passive (non-piezo), 3- Void. This form of multi-material TO lets the elements in the design domain switch between the passive material, active material or void during the sequence of optimization iterations. Using this form of multi-material TO, Kang et al. (2011) considered the voltage profile in the optimization problem to find an optimal layout for the voltage application. In the other form of multi-material TO, Gonçalves et al. (2018) considered two phases of active and passive materials without the void phase for the design domain. Cao et al. (2021) also considered two phase material, by doing isogeometric optimization on the functionally graded material while considering different objective functions and deriving the explicit formulation for the sensitivity analysis. The other form of multi-material approach is proposed by Kang and Tong (2008) and Kang et al. (2012) in which the passive and active materials are optimized simultaneously. However, there was no switch between active and passive materials. Indeed, the 3 phase multi-material TO with switch between active and passive materials is the most advanced and efficient format, while realization and fabrication may be less feasible due to complex geometry and electrode patterns.

The researches which have been mentioned so far for the application of TO to piezoelectricity used the density-based approach in particular the SIMP approach. However, other approaches are used as well. For example, Chen et al. (2010) applied the level set method in both 2D design domains as plates and 3D design domains as cylindrical shapes. Townsend et al. (2019) also used the level set method to optimize the PEH for structural health monitoring. In this research, a piezoelectric patch was attached to a passive layer and the level set method was used to optimize the eigenfrequency and eigenmodes of the PEH. de Almeida et al. (2019) employed the Bi-directional ESO (BESO) approach to optimize the topology of a cantilever PEH in the direction of thickness. The major contribution of this research was proposing a method for optimizing the placement of the electrodes on the top and bottom of the beam. Despite the utilization of other approaches, the density-based approach has been more dominant in the literature in terms of popularity.

TO is well established in the literature by publication of the implementation codes that helped researchers and newcomers to understand the methodology. In this case, Sigmund (2001a) published the first implementation code of TO in MATLAB software. In this paper, the SIMP approach was used to solve the structural compliance problems. several successors to this code were later published. For example, Andreassen et al. (2011) published 88 lines of MATLAB code which was faster than the 99 lines by Sigmund (2001a), thanks to the vectorization and introduction of the connectivity matrix for assembly of elemental matrices. Liu and Tovar (2014) published a 169 lines of code which was an extension of the 88 lines code for 3D TO. In this paper, the efficiency of the code is examined by solving the compliance problems, compliant mechanisms and heat transfer problems. Wang et al. (2021) did a review of all the educational implementation codes for TO in different physics and applications.

Inspired by the published implementation codes, some of the authors of the present paper published the first TO implementation codes for piezoelectric materials in 2D (Homayouni-Amlashi et al., 2021). In this paper, two MATLAB implementation codes were developed for actuation and energy harvesting. Using these codes, Yang et al. (2022) implemented their contribution on increasing the reliability of TO in application to piezoelectric actuator with voltage uncertainty and Schlinquer et al. (2020) optimized the topology and polarization profile of a single layer piezoelectric actuator working as a pusher. Since the employed finite element model (FEM) was in 2D, no force or deflection is considered out-of-plane. Despite the efficiency of the codes for in-plane applications, planar deformation and excitation were huge restrictions based on the real application of piezoelectric-based devices which usually work in out-of-plane motion. This motivated the authors to extend those published codes to 3D piezoelectric structures. Extension of TO from 2D to 3D in piezoelectric materials is more challenging than the similar extension for passive materials in compliance problems. Piezoelectric materials are not isotropic and the polarization direction of these materials affects the mechanical, coupling and dielectric tensor matrices. Moreover, piezoelectric materials in 3D are in the layer-based form where each layer is sandwiched between the electrodes. Indeed, the definition of electrodes is a challenging process that can affect the behavior of the multilayer piezoelectric plates and the result of optimization. Another challenge arises due to the fact that in many piezoelectric actuators, sensors or energy harvesters a combination of piezo and non-piezo materials is considered.

In this paper, two codes are provided to apply TO to multimaterial piezoelectric actuators and energy harvesters in a 3D design domain. The FEM is established by utilization of two types of 3D hexahedral elements: 1) 8 node trilinear elements, 2) 20 nodes quadratic elements. Trilinear elements have less precision especially in bending deformation while 20 nodes elements provide more precision at the expense of more computation time. Users can choose the element type at the beginning of the code. To provide proper criteria for choosing the type of the element, a study is provided in which the developed FEM in MATLAB is compared and verified by FEM simulations in COMSOL. Through this analysis, users will know when they can use the trilinear elements to reduce the computation time and when they are obliged to use the quadratic elements.

The codes are written in the most flexible format to have the capability of considering different case studies. By default, codes consider the existence of non-piezoelectric material in addition to the piezoelectric materials in the design domain. The optimization will be done on both materials at the same time but there will be no switch between the non-piezo and piezo materials. In this regard, the geometry of the non/piezo and piezo materials can be defined independently and the user is free to choose the desired resolution of the discretization mesh. Users can freely choose the desired materials for the piezo and non-piezo domains. In this case, the constitutive matrices for non-isotropic piezoelectric materials depend on the directions of polarization that can be freely chosen by the user as well. By default, three directions of polarization are considered in the code. However, with small modifications, one can define any direction of polarization. Users are free to consider any number of piezoelectric layers for multi-layer piezoelectric material. In this regard, the definition of electrodes can be done easily thanks to the proposed indexing method. This indexing method can easily index the elements, nodes and Degrees of Freedom (DOFs) which also facilitates the definition of forces and boundary conditions easier than what exists in the published codes in the literature. The indexing proposed here can also be integrated into other implementation codes that are already published in the literature. In addition to what is proposed in this paper, the codes are benefiting from the recent developments in terms of filtering, speed-up techniques and updating algorithm. For the filtering method, the projection suggested by Wang et al. (2011) is used beside the density filter by using the codes developed by Ferrari and Sigmund (2020) and the results are satisfying. The speed-up techniques which are also suggested by Ferrari and Sigmund (2020) are used in the codes. These speed-up techniques concern the assembly procedure. For the updating algorithm (optimizer), the code developed by Ferrari et al. (2021) is used as well. The performance of this optimizer is satisfactory for the educational purposes of this paper. Indeed, this updating algorithm lets the two developed codes be run independently without calling any external codes. However, as stated by Ferrari et al. (2021), it is not a fully stable optimizer and hence the readers are advised to use more established updating algorithms such as Method of Moving Asymptotes (MMA) (Svanberg, 1987, 2007) for more advanced studies.

The developed codes are accompanied by five add-on functions which will be used by both actuation and energy harvesting codes. The first add-on function is for the display of the result. This function is written to visualize the results in 3D-rendered form and also in 2D layer-based form which shows the density and polarization layout in each layer separately. The second add-on function is the updating algorithm developed by Ferrari et al. (2021). The third add-on function is for the calculation of the elemental stiffness matrices which works for both piezo and non-piezo materials. The fourth add-on function is for the rotation of the local coordinate system based on the direction of polarization which produces new tensor matrices. The last add-on function visualizes the deformation of the 3D structure with elemental representation based on the displacement vector.

The paper is organized as follows: in section 2, the FEM of the design domains consisting of piezo and non-piezo material is presented which includes rotation of polarization direction calculation of elemental and global stiffness matrices, providing the equilibrium equations, normalization and application of boundary conditions. In section 3, the theoretical aspects of the SIMP topology optimization with its extension (PEMAP-P) are presented without repeating what has been already presented in the previously published codes by Homayouni-Amlashi et al. (2021) and the addition of the non-piezo material is considered in the analytical part. In section 4, each part of the implementation codes is explained. The detailed explanation in this section facilitates the comprehension of the codes and their development for different case studies. In section 5, several numerical examples are provided to demonstrate the efficiency and flexibility of the developed codes in dealing with various case studies. The last section provides the discussion and conclusion.

2 Finite element modeling

In this section, the goal is to provide a Finite Element Model (FEM) for a design domain consisting of piezoelectric and non-piezoelectric materials. The piezoelectric material is defined as active and non-piezoelectric material as passive in the rest of the paper. Without loss of generality, several assumptions have been made for the modeling of the piezoelectric material:

- the electromechanical system is linear,
- the thickness of the electrodes is negligible and they are not modeled in the FEM,
- the electrodes are perfectly conductive,
- passive and active domains are set and fixed in FEM modeling and before the optimization. As such, passive and active domains remain passive and active respectively, in the sequence of optimization iterations.

The last assumption states that there is no switch between active and passive domains. In the upcoming section, the FEM of piezoelectric materials will be presented and the inclusion of passive material inside the design domain will be discussed.

2.1 Piezoelectric constitutive equation

Piezoelectricity is a phenomenon that couples the electrical and mechanical states of some materials. This phenomenon exists naturally in some materials as it was discovered primarily by the Curie brothers or it can be produced artificially in some other materials (Motzki and Seelecke, 2022). The coupling phenomenon means that the piezoelectric material exhibits a mechanical deformation due to electrical charges (direct effect) and also produces electrical charges when it is subjected to mechanical stimuli (inverse effect). This coupling phenomenon is represented by the governing constitutive equation. Considering a piezoelectric body in 3D space that occupies the volume V, four different forms of constitutive equations can be obtained (Mattiat, 2013). Each of these four forms is obtained based on the choice of the independent variables which are electric field ($\bar{E}$), charge density ($\bar{D}$), mechanical stress ($\bar{T}$) and mechanical strain ($\bar{S}$). For each of these four forms, two of these variables are considered independent and the rest are obtained accordingly through the constitutive equations. One of these forms known as "e form" or "stress-charge" is more convenient for the study in this paper, is more popular and can be stated as follows (Lerch, 1990),

$$\begin{aligned} \bar{T} &= c^E \bar{S} - e\bar{E} \\ \bar{D} &= e^T \bar{S} + \varepsilon^S \bar{E} \end{aligned} \tag{1}$$

In (1), c^E is the stiffness tensor in a constant electrical field, e is the piezoelectric matrix and ε^S is the matrix of permittivity in constant mechanical strain and T represents the matrix transpose.

The derivation of the aforementioned constitutive equations and the other 3 forms of these equations are mentioned in (Meeker, 1996; Tiersten, 2013). The elements of tensor matrices in the constitutive equation (1) depend on the type of the piezoelectric material (Mattiat, 2013; Piefort, 2001). However, the type of piezoelectric material will not affect the FEM procedure. Therefore, the following general format of the piezoelectric matrices will be considered

$$\begin{aligned} c^E &= \begin{bmatrix} c^E_{11} & c^E_{12} & c^E_{13} & c^E_{14} & 0 & 0 \\ c^E_{12} & c^E_{22} & c^E_{13} & c^E_{24} & 0 & 0 \\ c^E_{13} & c^E_{13} & c^E_{33} & 0 & 0 & 0 \\ c^E_{14} & c^E_{24} & 0 & c^E_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & c^E_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & c^E_{66} \end{bmatrix} \\ e^T &= \begin{bmatrix} 0 & 0 & 0 & 0 & e_{15} & e_{16} \\ e_{21} & e_{22} & & e_{24} & 0 & 0 \\ e_{31} & e_{32} & e_{33} & 0 & 0 & 0 \end{bmatrix} \\ \varepsilon^S &= \begin{bmatrix} \varepsilon^S_{11} & 0 & 0 \\ 0 & \varepsilon^S_{22} & 0 \\ 0 & 0 & \varepsilon^S_{33} \end{bmatrix} \end{aligned} \tag{2}$$

The tensor matrices in (2), include PZT (lead zirconate titanate) materials with 4mm tetragonal crystal class (Piefort, 2001), Lithium Niobate (LiNBO3) with trigonal crystal class (Weis and Gaylord, 1985), Aluminum Nitride (AlN) as a ceramic material and Polyvinylidene Fluoride (PVDF) as a polymer (Kalimuldina et al., 2020). The corresponding coefficients of the tensor matrices for these materials are reported in the appendix. By default, PZT-4 is chosen as the piezoelectric material in the paper and any change will be announced where it is necessary. The change in the piezoelectric material will change the result of optimization in general. However, users can freely choose any piezoelectric material even those whose tensor matrices do not follow the ones in equation (2), without the need to modify the implementation codes.

No matter the type of piezoelectric material, the polarization direction modifies the tensor matrices which will be discussed next.

2.2 Rotation of polarization direction

The polarization direction is the direction in which the dipoles inside the piezoelectric materials are aligned through the polling process. In this process, a strong electrical field will be applied to the piezoelectric material in a particular direction which makes unidirectional dipoles in the piezoelectric materials (Sekhar et al., 2021). In general, the data sheets for piezoelectric materials provide the piezoelectric coefficients for the case where the polarization direction is in the $\vec{z}$ direction of the coordinate axes. Here, we consider the cases where the polarization direction is in other coordinate directions. By rotating the coordinate axes, the tensor matrices which are mentioned in (2) should be modified. This modification will be done through the rotation of the local coordinates. The transformation of the tensor matrices for general anisotropic material due to the rotation of the local coordinates is addressed by Fahmy and Adler (1975). The rotation of matrices can be done using the Euler angles as illustrated in Fig. 1. The Euler angles are specified as α, β and γ. If the tensor matrices of the piezoelectric material in the $\vec{x}$, $\vec{y}$, $\vec{z}$ coordinates are known, then the tensor matrices in the new coordinate system for which the rotation is specified by Euler angles can be derived as follows (Fahmy and Adler, 1975)

$$
\begin{aligned}
C_{new} &= Zc^E Z^T \\
e_{new} &= LeZ^T \\
\varepsilon_{new} &= L\varepsilon^S L^T
\end{aligned}
\tag{3}
$$

In which, c^E, e and ε^S are the piezoelectric tensor matrices as they are introduced in equation (2) while L and Z are the rotation matrices. The derivation of these rotation matrices is as follows (Fahmy and Adler, 1975)

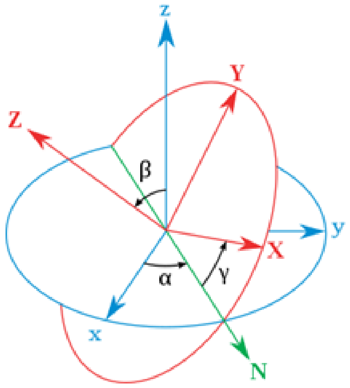

Fig. 1 Euler angles

$$
L = \begin{bmatrix} \chi_1 & \Theta_1 & \Psi_1 \\ \chi_2 & \Theta_2 & \Psi_2 \\ \chi_3 & \Theta_3 & \Psi_3 \end{bmatrix} \tag{4}
$$

$$
Z = \begin{bmatrix}
\chi_1^2 & \Theta_1^2 & \Psi_1^2 & 2\Theta_1\chi_1 & 2\gamma_1\chi_1 & 2\alpha_1\beta_1 \\
\chi_2^2 & \Theta_2^2 & \Psi_2^2 & 2\Theta_2\chi_2 & 2\gamma_2\chi_2 & 2\alpha_2\beta_2 \\
\chi_3^2 & \Theta_3^2 & \Psi_3^2 & 2\Theta_3\chi_3 & 2\gamma_3\chi_3 & 2\alpha_3\beta_3 \\
\chi_2\chi_3 & \Theta_2\Theta_3 & \Psi_2\Psi_3 & \Theta_2\Psi_3+\Theta_3\Psi_2 & \Psi_2\chi_3+\Psi_3\chi_2 & \chi_2\Theta_3+\chi_3\Theta_2 \\
\chi_3\chi_1 & \Theta_3\Theta_1 & \Psi_3\Psi_1 & \Theta_1\Psi_3+\Theta_3\Psi_2 & \Psi_1\chi_3+\Psi_3\chi_2 & \chi_1\Theta_3+\chi_3\Theta_2 \\
\chi_1\chi_2 & \Theta_1\Theta_2 & \Psi_1\Psi_2 & \Theta_1\Psi_2+\Theta_2\Psi_1 & \Psi_1\chi_2+\Psi_2\chi_1 & \chi_1\Theta_2+\chi_2\Theta_1
\end{bmatrix} \tag{5}
$$

where the variables χ, Θ and Ψ can be derived based on the Euler angles

$$
\begin{aligned}
\chi_1 &= cos(\gamma)cos(\alpha) - cos(\beta)sin(\alpha)sin(\gamma) \\
\chi_2 &= -sin(\gamma)cos(\alpha) - cos(\beta)sin(\alpha)cos(\gamma) \\
\chi_3 &= sin(\beta)sin(\alpha) \\
\Theta_1 &= cos(\gamma)sin(\alpha) + cos(\beta)cos(\alpha)sin(\gamma) \\
\Theta_2 &= -sin(\gamma)sin(\alpha) + cos(\beta)cos(\alpha)cos(\gamma) \\
\Theta_3 &= -sin(\beta)cos(\alpha) \\
\Psi_1 &= sin(\gamma)sin(\beta) \\
\Psi_2 &= cos(\alpha)sin(\beta) \\
\Psi_3 &= cos(\beta)
\end{aligned}
\tag{6}
$$

By knowing the necessary Euler angles, the rotation matrices (L and Z) can be derived. Therefore, the only remaining point is the definition of the rotation for Euler angles to rotate the coordinates system. It is known that three composed rotations of the axes are enough to reach any target frame and there are different conventions for the sequence of the axes to define the Euler angles. The sequence of the axes which is used by

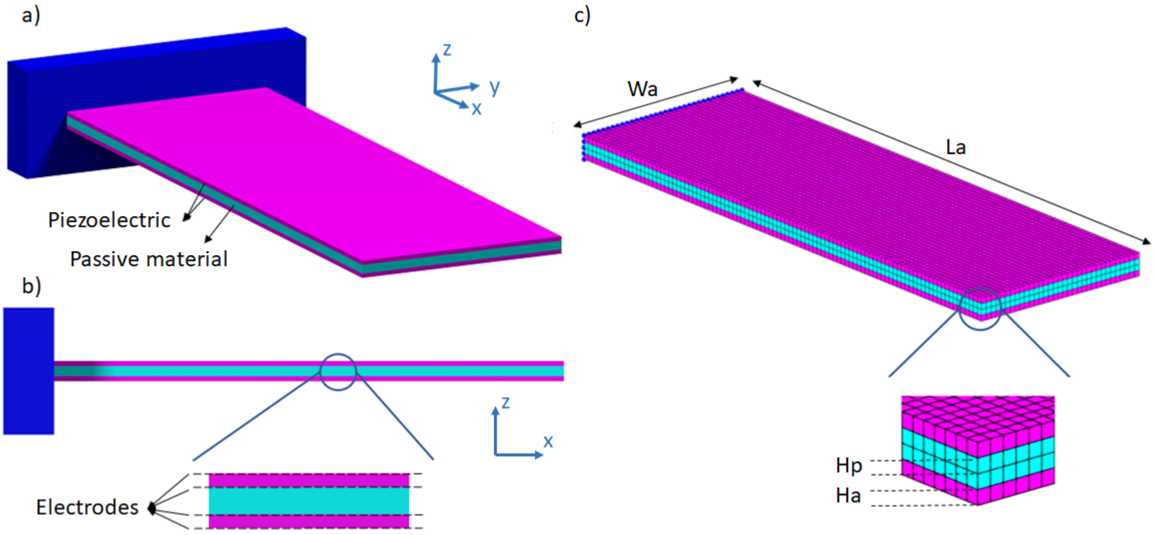

Fig. 2 Multi-layer piezoelectric plate consisting active piezoelectric and passive material. a) Isometric view, b) Side view, c) Discretization of design domain to the finite number of elements, La = length of active material, Wa = width of active material, Ha = element-wise thickness of active layer.

Fahmy and Adler (1975) to define the Euler angles are $\vec{z}$, $\vec{x}$, $\vec{z}$. This means the first rotation will be around the $\vec{z}$ axis which is shown by angle α. The second rotation around the new $\vec{x}$ axis will be shown as β and the last rotation around the new $\vec{z}$ axis is specified by γ. By having these rotation angles and by using the equations (3)-(6), the new tensor matrices can be derived. In this case, any polarization direction can be considered in the formulation and the rotation from the coordinates axis of the data sheet to the desired one can be done through definitions of Euler angles and rotation matrices.

2.3 3D finite element model of piezoelectric material

After establishing the constitutive equations, the next step is to build the FEM of the piezoelectric material. We can consider a design domain, consisting of active and passive material as it has been shown in Fig. (2). In this figure, a passive layer that is sandwiched between two piezoelectric layers has been depicted. This is the most common configuration of passive and active piezoelectric material in the literature due to its effectiveness and simplicity of fabrication. The proposed FEM, optimization and codes are general enough to consider various configurations. However, the configuration in Fig. (2) is chosen as a general candidate to demonstrate a 3D domain consisting of passive and active material. Each piezoelectric layer in the design domain is sandwiched between two electrodes on top and bottom (Fig. 2-(b)). The electrodes are considered thin enough to be neglected in the FEM. The polarization axis for the piezoelectric materials is parallel to the $\vec{z}$ axis of the coordinate system while the direction will be later defined by the optimization method. The FEM starts by discretizing the design domain using a finite number of 3D rectangular elements as shown in Fig. 2-(c). 3D rectangular element is a particular form of hexahedral elements (Hutton and Wu, 2004). Here, we consider two types of hexahedral elements. The first type is the trilinear 8 nodes element which has been used by Zheng et al. (2009), Noh and Yoon (2012) and Homayouni-Amlashi et al. (2019) for 3D TO of piezoelectric energy harvesters. This is the same type of element which has been used by Liu and Tovar (2014) for developing the 3D topology optimization MATLAB codes for passive materials. Although these elements are used for TO of piezoelectric materials, they have a major defect known as shear locking (Cook et al., 2007). As such, trilinear elements can not exhibit pure bending due to existence of a parasitic shear strain. As a second type, we consider 20 nodes quadratic elements which are more precise than the trilinear 8 nodes elements and don't exhibit shear locking and parasitic shear strain. However, in terms of computation time trilinear elements are more economical. In the appendix, an analysis is performed to investigate the accuracy of both elements in bending piezoelectric actuators. Based on this analysis, the thickness to length ratio of each piezoelectric layer determines whether we should use trilinear or quadratic elements. For thickness to length ratio below 0.05, we have to use the quadratic elements since the error in trilinear elements is enormous. For higher thickness to length ratios, utilization of trilinear elements is possible with recommendation of using at least two elements per thickness of each layer. To have a better understanding of the criteria for choosing the type of the element readers are advised to study the appendix.

From now on, independent of the type of the element, we call the elements that belong to active material as active elements and the rest are considered to be passive elements. By applying Hamilton's variational principle to the piezoelectric constitutive equation which is explained in detail by Lerch (1990) and Piefort (2001), the linear finite element differential equation is derived for the design domain. This linear finite element equation for the actuation case in which the static analysis is considered can be written as (Homayouni-Amlashi et al., 2021),

$$K_{uu}U + K_{u\phi}\Phi = F \tag{7}$$

Where K_{uu} and $K_{u\phi}$ are the global stiffness and coupling matrices after the assembly of all active and passive elements. U and Φ are the global vectors of mechanical displacements and potentials respectively. For the energy harvesting case, the global FEM equation considering the harmonic excitation while neglecting the damping effect can be found as (Homayouni-Amlashi et al., 2021),

$$\begin{bmatrix} K_{uu} - M\Omega^2 & K_{u\phi} \\ K_{\phi u} & -K_{\phi\phi} \end{bmatrix} \begin{bmatrix} U \\ \Phi \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix} \tag{8}$$

where $K_{\phi\phi}$ is the global dielectric stiffness matrix and Ω is the frequency of excitation. The detailed explanation of the derivation of equations (7) and (8) has been mentioned in (Homayouni-Amlashi et al., 2021). For the assembly, active and passive elements are assembled in a regular procedure taking into consideration that the elemental coupling matrix ($k_{u\phi}$) and dielectric stiffness matrix ($k_{\phi\phi}$) are null matrices for passive elements. The assembly procedure of passive and active elements inside the design domain will be discussed in section 4. After the assembly, to solve the FEM equations (7) and (8), the mechanical boundary condition and equipotential condition should be applied. This will be explained in detail in section 4.

After deriving the global equilibrium finite element equations, we have to formulate the elemental stifness matrices. Depending on the type of element, the formulation of the elemental stiffness matrices will be different. Therefore, calculation of trilinear elements and quadratic elements will be given in the following two separate subsections.

2.3.1 Trilinear 8 node elements

Trilinear hexahedral elements have 8 nodes and each node has three mechanical DOFs as displacement and one electrical DOF in the case of piezoelectric material (Lerch, 1990). In Fig. 3, a schematic coarse discretization of the design domain is illustrated with the numbering pattern of these elements, nodes and mechanical DOFs. The elements are numbered from top to bottom and from back to front. The same numbering pattern is followed to number the nodes. To specify the mechanical DOF for each node, it is enough to multiply the node number by three and consider the two previous sequences of numbers. It is important to note that, the internal numbering of nodes in each element is different from the external numbering pattern. That is why in panel (c) of Fig. 3, the internal numbering sequence of nodes does not follow the external numbering of nodes in panel (b). For active elements, due to the electromechanical coupling effect of the piezoelectricity, there will be one electrical DOF as potential per node while there is no electrical DOF for passive elements. The 3D rectangular elements can have arbitrary length, thickness and height. This will provide the flexibility in definition of the dimensions of the design domain and the definition of mesh resolution independently.

The procedure for calculating the elemental matrices for 3D elements is similar to the procedure for 2D elements (Homayouni-Amlashi et al., 2021) with some modifications. Moreover, this procedure remains the same for passive and active elements and the difference will be announced where it is necessary. First of all, the displacement of each point in the element based on global coordinate can be expressed by the displacement of the nodes as follows (Hutton and Wu, 2004)

$$\begin{aligned} x &= n_1x_1 + n_2x_2 + n_3x_3 + n_4x_4 + n_5x_5 + \\ &n_6x_6 + n_7x_7 + n_8x_8 \\ y &= n_1y_1 + n_2y_2 + n_3y_3 + n_4y_4 + n_5y_5 + \\ &n_6y_6 + n_7y_7 + n_8y_8 \\ z &= n_1z_1 + n_2z_2 + n_3z_3 + n_4z_4 + n_5z_5 + \\ &n_6z_6 + n_7z_7 + n_8z_8 \end{aligned} \tag{9}$$

where x_i, y_i, z_i are the global coordinates and n_i are the interpolation functions that are defined as

$$N = \frac{(1+\xi\xi_i)(1+\eta\eta_i)(1+\mu\mu_i)}{8} \tag{10}$$

where ξ, η, μ are the natural coordinates as have been shown in Fig. 3-(c) and ξ_i, η_i, μ_i are the coordinates of

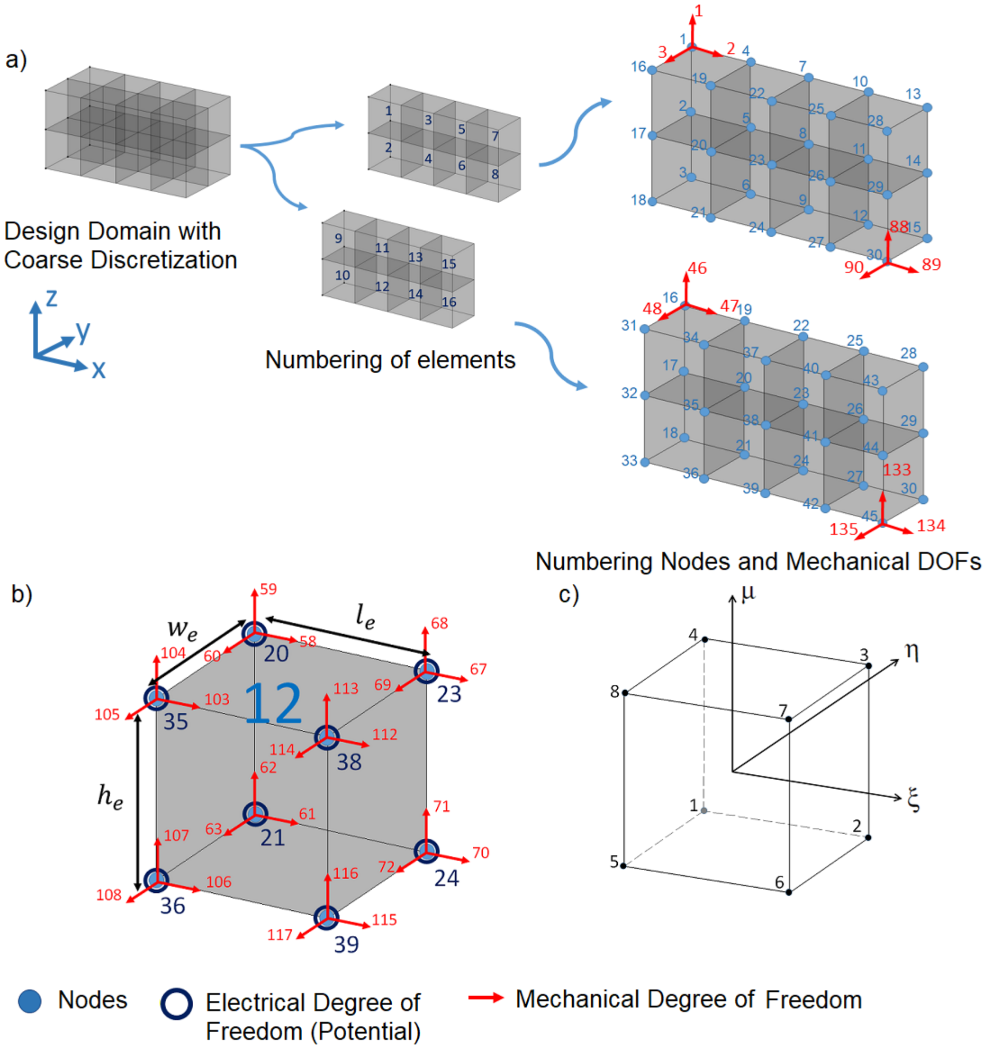
a)
Design Domain with Coarse Discretization
z
y
x
Numbering of elements
Numbering Nodes and Mechanical DOFs
b)
w_e
l_e
h_e
12
20
23
35
38
21
24
36
39
c)
μ
η
ξ
1
2
3
4
5
6
7
8
Nodes
Electrical Degree of Freedom (Potential)
Mechanical Degree of Freedom

Fig. 3 Discretization of design domain with 3D trilinear hexahedral elements. a) Numbering pattern in global format, b) Numbering pattern inside the element c) Parent element in natural coordinate.

the nodes in parent element as illustrated in Fig. 3-(c). In this figure, for the node number (1) we have $\xi_i, \eta_i, \mu_i = -1$ and for node number (7) we have $\xi_i, \eta_i, \mu_i = 1$. The world "trilinear" is chosen since each interpolation function contains the product of three linear functions (Cook et al., 2007). For each element, the strain and the electrical field can be calculated based on the mechanical displacement and electrical field,

$$\bar{S} = B_u u, \qquad \bar{E} = B_\phi \phi \tag{11}$$

where u is the vector of all the mechanical displacements inside each element and ϕ is the potential vector containing the potential values of each node inside the active element. The matrices B_u and B_ϕ are called gradient interpolation matrices while the former is also called the strain-displacement matrix that can be defined as

$$B_u = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{bmatrix} N \tag{12}$$

where N is the matrix of interpolation functions

$$N = \begin{bmatrix} n_1 & 0 & 0 & n_2 & 0 & 0 & \cdots & n_8 & 0 & 0 \\ 0 & n_1 & 0 & 0 & n_2 & 0 & \cdots & 0 & n_8 & 0 \\ 0 & 0 & n_1 & 0 & 0 & n_2 & \cdots & 0 & 0 & n_8 \end{bmatrix} \tag{13}$$

Since the derivative of the interpolation functions with respect to global coordinates is required, an indirect approach through using the Jacobian matrix can be used (Hutton and Wu, 2004). Since we have the derivative of the interpolation functions with respect to the natural coordinates, we can write

$$\begin{Bmatrix} \frac{\partial n_i}{\partial \xi} \\ \frac{\partial n_i}{\partial \eta} \\ \frac{\partial n_i}{\partial \mu} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \mu} & \frac{\partial y}{\partial \mu} & \frac{\partial z}{\partial \mu} \end{bmatrix} \begin{Bmatrix} \frac{\partial n_i}{\partial x} \\ \frac{\partial n_i}{\partial y} \\ \frac{\partial n_i}{\partial z} \end{Bmatrix} \qquad i = 1:8 \tag{14}$$

where the Jacobian matrix can be identified as

$$J = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \mu} & \frac{\partial y}{\partial \mu} & \frac{\partial z}{\partial \mu} \end{bmatrix} \tag{15}$$

and the derivative of the interpolation function with respect to the global coordinates can be obtained by

$$\begin{Bmatrix} \frac{\partial n_i}{\partial x} \\ \frac{\partial n_i}{\partial y} \\ \frac{\partial n_i}{\partial z} \end{Bmatrix} = J^{-1} \begin{Bmatrix} \frac{\partial n_i}{\partial \xi} \\ \frac{\partial n_i}{\partial \eta} \\ \frac{\partial n_i}{\partial \mu} \end{Bmatrix} \tag{16}$$

Now, based on equation (12), the strain-displacement matrix is

$$B_u = J^{-1} \begin{bmatrix} \frac{\partial n_1}{\partial \xi} & 0 & 0 & \frac{\partial n_2}{\partial \xi} & 0 & 0 & \cdots & 0 \\ 0 & \frac{\partial n_1}{\partial \eta} & 0 & 0 & \frac{\partial n_2}{\partial \eta} & 0 & \cdots & 0 \\ 0 & 0 & \frac{\partial n_1}{\partial \mu} & 0 & 0 & \frac{\partial n_2}{\partial \mu} & \dots & \frac{\partial n_8}{\partial \mu} \\ 0 & \frac{\partial n_1}{\partial \mu} & \frac{\partial n_1}{\partial \eta} & 0 & \frac{\partial n_2}{\partial \mu} & \frac{\partial n_2}{\partial \eta} & \dots & \frac{\partial n_8}{\partial \eta} \\ \frac{\partial n_1}{\partial \mu} & 0 & \frac{\partial n_1}{\partial \xi} & \frac{\partial n_2}{\partial \mu} & 0 & \frac{\partial n_2}{\partial \xi} & \dots & \frac{\partial n_8}{\partial \xi} \\ \frac{\partial n_1}{\partial \eta} & \frac{\partial n_1}{\partial \xi} & 0 & \frac{\partial n_2}{\partial \eta} & \frac{\partial n_2}{\partial \xi} & 0 & \cdots & 0 \end{bmatrix} \tag{17}$$

It is obvious that we have 3 defined displacements for each node as mechanical DOF. For active elements the gradient interpolation matrix (B_ϕ) which relates the potential to the electrical field is obtained by

$$B_\phi = J^{-1} \left[\frac{\partial N}{\partial x} \; \frac{\partial N}{\partial y} \; \frac{\partial N}{\partial z} \right]^T =$$
$$\begin{bmatrix} B_{\phi,11} & B_{\phi,12} & B_{\phi,13} & B_{\phi,14} & B_{\phi,15} & B_{\phi,16} & B_{\phi,17} & B_{\phi,18} \\ B_{\phi,21} & B_{\phi,22} & B_{\phi,23} & B_{\phi,24} & B_{\phi,25} & B_{\phi,26} & B_{\phi,27} & B_{\phi,28} \\ B_{\phi,31} & B_{\phi,32} & B_{\phi,33} & B_{\phi,34} & B_{\phi,35} & B_{\phi,36} & B_{\phi,37} & B_{\phi,38} \end{bmatrix} \tag{18}$$

With the help of the Matrix B_ϕ it is possible to rewrite the strain displacement matrix as follows,

$$B_u = \begin{bmatrix} B_{\phi,11} & 0 & 0 & B_{\phi,12} & 0 & 0 & \cdots & 0 \\ 0 & B_{\phi,21} & 0 & 0 & B_{\phi,22} & 0 & \cdots & 0 \\ 0 & 0 & B_{\phi,31} & 0 & 0 & B_{\phi,32} & \cdots & B_{\phi,38} \\ 0 & B_{\phi,31} & B_{\phi,21} & 0 & B_{\phi,32} & B_{\phi,22} & \cdots & B_{\phi,28} \\ B_{\phi,31} & 0 & B_{\phi,11} & B_{\phi,32} & 0 & B_{\phi,12} & \cdots & B_{\phi,18} \\ B_{\phi,21} & & 0 & B_{\phi,22} & B_{\phi,12} & 0 & \cdots & 0 \end{bmatrix} \tag{19}$$

Indeed, in the calculation of the strain displacement matrix for passive materials, the calculation of matrix B_ϕ was an intermediary step. However, for the piezoelectric material that intermediary step gives us the potential to electrical displacement matrix known as B_ϕ. Finally, the elemental matrices can be calculated (Hutton and Wu, 2004; Lerch, 1990)

$$\begin{aligned} k_{uu} &= \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} B_u^T c^E B_u |J| d\xi d\eta d\mu \\ k_{u\phi} &= \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} B_u^T e^T B_\phi |J| d\xi d\eta d\mu \\ k_{\phi\phi} &= \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} B_\phi^T \varepsilon B_\phi |J| d\xi d\eta d\mu \\ m &= \rho \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} N^T N |J| d\xi d\eta d\mu \end{aligned} \tag{20}$$

After the calculation of the elemental matrices, the global matrices can be assembled with the numbering format which is illustrated in Fig. 3. Then, the global FEM equation of motion can be written for actuation and energy harvesting cases separately.

2.3.2 Quadratic 20 node elements

In Fig. 4, the coarse discretization of the design domain with quadratic element is depicted. The numbering of elements is following the same order as trilinear elements. For the numbering of the nodes, an ordering is proposed in which the code can be flexible in terms of switching between the types of elements. In this ordering format, first all the corner nodes are numbered in a similar manner to trilinear elements. Therefore, the numbering format in panel (a) is the same as trilinear elements. After numbering the corner nodes, midside nodes are numbered as illustrated in panels (b-d). To better understand the numbering format, element number (6) is chosen and all the node numbers are illustrated in panel (e). The node numbering format in the parent element is also demonstrated in panel (f).

The procedure for calculation of the elemental matrices for quadratic elements is the same as trilinear elements. The difference lies in the existence of additional nodes. For example, equation (9) should be modified as

$$x = n_1x_1 + n_2x_2 + n_3x_3 + n_4x_4 + n_5x_5+$$
$$n_6x_6 + n_7x_7 + n_8x_8 + ... + n_{20}x_{20}$$
$$y = n_1y_1 + n_2y_2 + n_3y_3 + n_4y_4 + n_5y_5+$$
$$n_6y_6 + n_7y_7 + n_8y_8 + ... + n_{20}y_{20}$$
$$z = n_1z_1 + n_2z_2 + n_3z_3 + n_4z_4 + n_5z_5+$$
$$n_6z_6 + n_7z_7 + n_8z_8 + ... + n_{20}z_{20} \tag{21}$$

The interpolation functions for quadratic elements are mentioned in (Logan, 2007). These interpolation functions can be expressed based on the coordinates of the nodes. For the corner nodes where we we have $\xi_i, \eta_i, \mu_i, = \pm 1$, the interpolation function can be written as

$$n_i =$$
$$(1/8)(1 + \xi\xi_i)(1 + \eta\eta_i)(1 + \mu\mu_i)(\xi\xi_i + \eta\eta_i + \mu\mu_i - 2) \tag{22}$$

Equation (22) provides the interpolation functions for the nodes at eighth corners. For the midside nodes at $\xi_i = 0, \eta_i = \pm 1, \mu_i = \pm 1$, the interpolation functions are expressed as

$$n_i = (1/4)(1 - \xi^2)(1 + \eta\eta_i)(1 + \mu\mu_i) \tag{23}$$

For the midside nodes at $\xi_i = \pm 1, \eta_i = 0, \mu_i = \pm 1$, the interpolation functions are

$$n_i = (1/4)(1 + \xi\xi_i)(1 - \eta^2)(1 + \mu\mu_i) \tag{24}$$

and finally for the midside nodes at $\xi_i = \pm 1, \eta_i = \pm 1, \mu_i = 0$, the interpolation functions are

$$n_i = (1/4)(1 + \xi\xi_i)(1 + \eta\eta_i)(1 - \mu^2) \tag{25}$$

Equations (22)-(25) provide 20 interpolation functions. Having these interpolation functions it is possible to calculate the elemental matrices following the same procedure for trilinear elements which is explained in section 4.

So far, a 3D finite element model of a body consisting of piezoelectric material and passive material has been built by discretizing the design domain with 3D hexahedral elements and derivation of the FEM equation. Now, it is possible to solve the FEM equation and formulate the optimization problem. Beforehand, a normalization of the governing equation will be presented to avoid numerical instabilities due to scale differences between piezoelectric tensor matrices.

2.4 Normalization

A normalization procedure for the global FEM equation for both actuation and energy harvesting was already established in (Homayouni-Amlashi et al., 2021). Since the format of the global FEM equations for the 2D and 3D FEM remains the same, the same normalization procedure can be utilized. The normalization starts by factorizing the highest value of each elemental matrix,

$$\tilde{k}_{uu} = k_{uu}/k_0, \quad \tilde{k}_{u\phi} = k_{u\phi}/\alpha_0$$
$$\tilde{k}_{\phi\phi} = k_{\phi\phi}/\beta_0, \quad \tilde{m} = m/m_0 \tag{26}$$

where $k_0, \alpha_0, \beta_0, m_0$ are the highest values of the corresponding matrices. Then, the new FEM equation for the actuation (7), can be written as

$$\tilde{K}_{uu}\tilde{U} + \tilde{K}_{u\phi}\tilde{\Phi} = \tilde{F} \tag{27}$$

in which

$$\tilde{F} = F/f_0, \quad \tilde{U} = U/u_0, \quad \tilde{\Phi} = \Phi/\phi_0$$
$$u_0 = f_0/k_0, \quad \phi_0 = f_0/\alpha_0 \tag{28}$$

and the new FEM equation for energy harvesting is derived as

$$\begin{bmatrix} \tilde{K}_{uu} - \tilde{M}\tilde{\Omega}^2 & \tilde{K}_{u\phi} \\ \tilde{K}_{\phi u} & -\gamma\tilde{K}_{\phi\phi} \end{bmatrix} \begin{bmatrix} \tilde{U} \\ \tilde{\Phi} \end{bmatrix} = \begin{bmatrix} \tilde{F} \\ 0 \end{bmatrix} \tag{29}$$

where

$$\tilde{\Omega}^2 = \Omega^2 m_0/k_0, \quad \gamma = k_0\beta_0/\alpha_0^2 \tag{30}$$

The proof of this normalization can be found in (Homayouni-Amlashi et al., 2021). After solving the FEM, we need to roll back the normalization and calculate the real outputs of the system (i.e. ϕ and U). In actuation mode, the input to the system is the potential and hence the value of Φ_0 is selected by the user a priory. As such, the real value of displacement can be calculated by

$$U = U_0\tilde{U} = \Phi_0\alpha_0\tilde{U}/k_0 \tag{31}$$

In the energy harvesting case, the force is the input and the value of f_0 is selected by the user a priory. Therefore, the real value of the potential can be calculated by

$$\Phi = \Phi_0\tilde{\Phi} = f_0\Phi/\alpha_0 \tag{32}$$

After the normalization, to solve the global FEM equation, the boundary condition should be applied which will be discussed next.

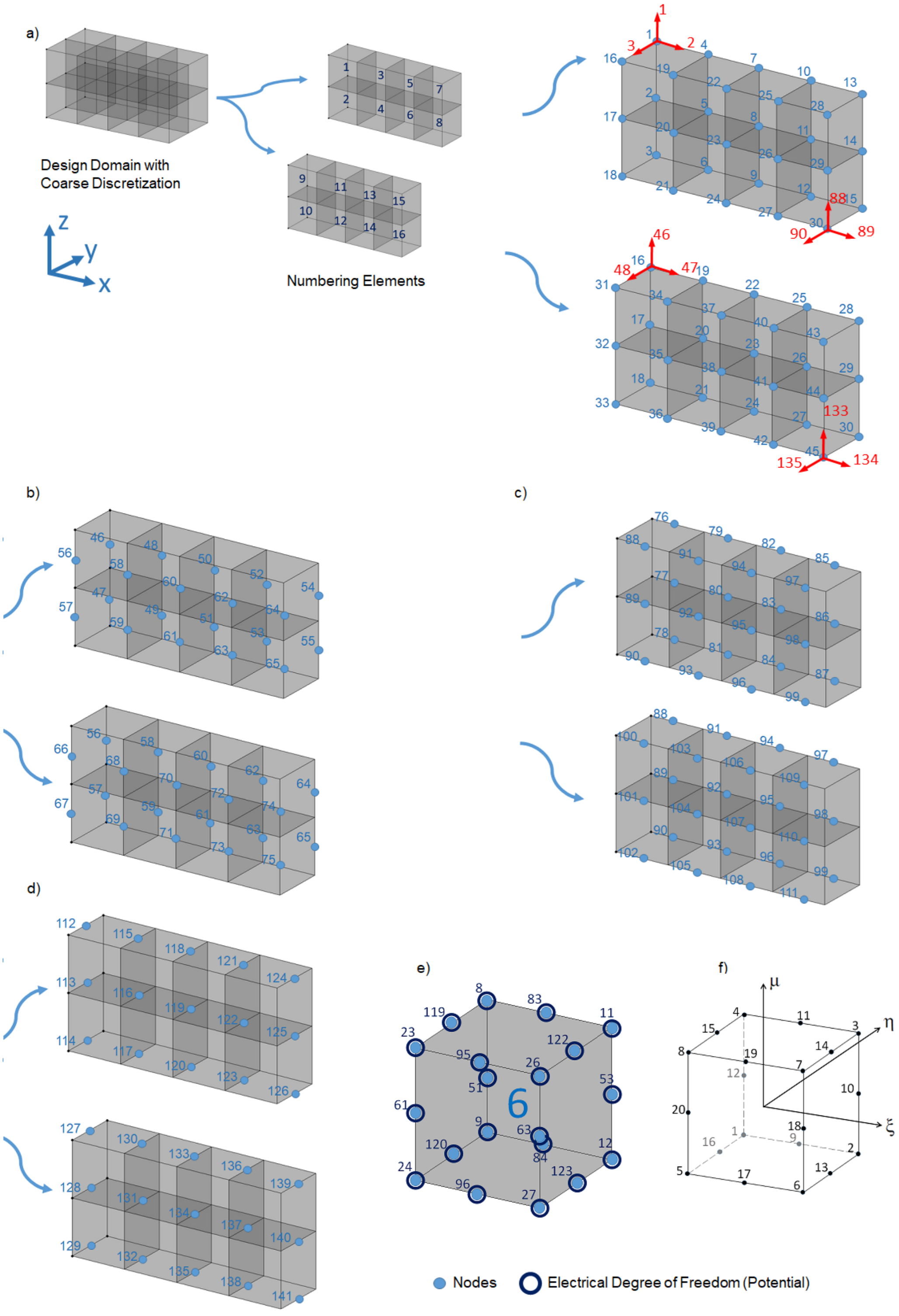

Fig. 4 Discretization of design domain with 3D quadratic hexahedral elements. a-d) Nodes numbering format, e) Numbering pattern inside the element f) Parent element in natural coordinate.

2.5 Boundary Conditions

Equations (27) and (29) provide the main global equilibrium equations for the actuation (in static mode) and energy harvesting after normalization. To solve these equations, mechanical and electrical boundary conditions should be applied. The application of the mechanical boundary conditions is similar to the compliance problems and it will be discussed in section 4. For the electrical boundary condition, different types of nodes should be identified. To do so, consider piezoelectric plates as they are illustrated in Fig. 5. We have 2 electrodes on top and bottom and the middle electrodes that we may consider as grounded (null potential). There is an equipotential condition on the face of the electrodes which are considered to be perfectly conductive. This equipotential condition will be applied in the following form (Homayouni-Amlashi et al., 2019)

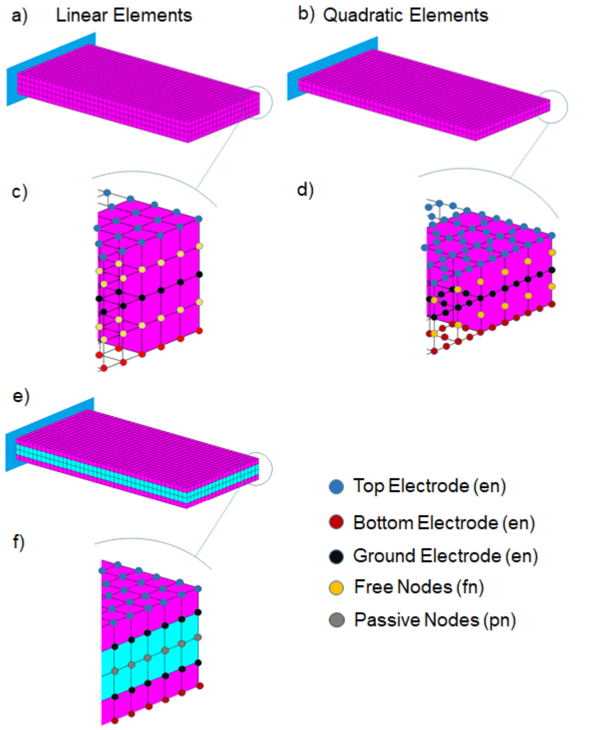

Fig. 5 Identifying different types of nodes for electrical boundary condition. en: equipotential nodes, a) 4-layer piezoelectric plate discretized by linear elements, b) 2-layer piezoelectric plate discretized by quadratic elements, c-d) Identification of different types of nodes, 4-layer plate consisting of piezoelectric and passive materials discretized by linear elements, f) Identification of different types of nodes

$$\Phi = BV_p \tag{33}$$

in which V_p is the vector of voltages for electrodes. In multi-layer piezoelectric plates, Boolean matrix B has the dimension of $N_n \times N_P$ where N_P is the number of potential electrodes and N_n is the number of nodes connected to electrodes. In this case, we have $N_P = 2$. In panel (a), there will be some nodes that are not connected to any electrodes as can be seen in Fig. 5-(c). These nodes are specified as free nodes (fn) while the nodes that are connected to electrodes are specified as equipotential nodes (en). For quadratic elements in panel (b) of Fig. 5, even by considering one element per thickness of each piezoelectric layer, the midside nodes are not connected to any electrodes and are specified as free nodes. Moreover, there are some nodes that belong to passive elements as illustrated in panel Fig. 5-(e). If the nodes of passive elements are not shared with active elements, they are considered as passive nodes (pn) with null potential as ground electrode.

The existence of free nodes in addition to potential nodes modifies the FEM modeling as reported in (Becker et al., 2006; Moretti and Silva, 2019). The modification can be applied to normalized equation (29) as

$$\begin{bmatrix} \tilde{K}_{uu} - \tilde{M}\tilde{\Omega}^2 & \tilde{K}_{u\phi_{fn}} & \tilde{K}_{u\phi_{en}} \\ \tilde{K}_{\phi_{fn}u} & -\gamma\tilde{K}_{\phi_{fn}\phi_{fn}} & -\gamma\tilde{K}_{\phi_{fr}\phi_{en}} \\ \tilde{K}_{\phi_{en}u} & -\gamma\tilde{K}_{\phi_{en}\phi_{fn}} & -\gamma\tilde{K}_{\phi_{pr}\phi_{en}} \end{bmatrix} \begin{bmatrix} \tilde{U} \\ \tilde{\Phi}_{fn} \\ \tilde{\Phi}_{pn} \end{bmatrix} = \begin{bmatrix} \tilde{F} \\ 0 \\ 0 \end{bmatrix} \tag{34}$$

where ϕ_{fn} and ϕ_{en} are the potentials of free nodes and equipotential nodes respectively. We eliminate the potentials of passive and grounded nodes as by removing the respective lines and columns in the FEM equation. By applying the equipotential condition (33), equation (34) is modified as

$$\begin{bmatrix} \tilde{K}_{uu} - \tilde{M}\tilde{\Omega}^2 & \tilde{K}_{u\phi_{fn}} & \tilde{K}_{u\phi_{en}}B \\ \tilde{K}_{\phi_{fn}u} & -\gamma\tilde{K}_{\phi_{fn}\phi_{fn}} & -\gamma\tilde{K}_{\phi_{fn}\phi_{en}}B \\ B^T\tilde{K}_{\phi_{en}u} & -\gamma B^T\tilde{K}_{\phi_{en}\phi_{fn}} & -\gamma B^T\tilde{K}_{\phi_{en}\phi_{en}}B \end{bmatrix} \begin{bmatrix} \tilde{U} \\ \tilde{\Phi}_{fn} \\ V_p \end{bmatrix} = \begin{bmatrix} \tilde{F} \\ 0 \\ 0 \end{bmatrix} \tag{35}$$

The FEM equation (35) is for the energy harvesting case. This equation can be written in abbreviate form as

$$[S_{EH}] \begin{bmatrix} \tilde{U} \\ \tilde{\Phi}_{fn} \\ V_p \end{bmatrix} = \begin{bmatrix} \tilde{F} \\ 0 \\ 0 \end{bmatrix} \tag{36}$$

In which, $[S_{EH}]$ is introduced for the brevity of later usage. For actuation, the FEM equation can be written as

$$\begin{bmatrix} \tilde{K}_{uu} & \tilde{K}_{u\phi_{fn}} \\ \tilde{K}_{\phi_{fn}u} & -\gamma\tilde{K}_{\phi_{fn}\phi_{fn}} \end{bmatrix} \begin{bmatrix} \tilde{U} \\ \tilde{\Phi}_{fn} \end{bmatrix} = \begin{bmatrix} \tilde{F} - \tilde{K}_{u\phi_{en}}\tilde{\Phi}_{en} \\ \gamma\tilde{K}_{\phi_{fn}\phi_{en}}\tilde{\Phi}_{en} \end{bmatrix} \tag{37}$$

For the actuation, since the prescribed voltages are known a periory, it is not necessary to apply the equipotential condition (33) since all the prescribed voltages will have the same values.

By having the final form of normalized FEM equations (35) and (37) for energy harvesting and actuation, we can enter the optimization phase which will be discussed in the next section.

3 Topology Optimization

To formulate the topology optimization problem for the piezoelectric actuator and energy harvester, the density-based approach and in particular the SIMP approach is employed in this paper. In this regard, the main step is to provide the material interpolation scheme for active and passive materials which will be discussed next.

3.1 Material interpolation scheme

The material interpolation scheme for the piezoelectric material is the extension of the SIMP approach known as PEMAP-P proposed by Kögl and Silva (2005). This interpolation scheme is implemented in (Homayouni-Amlashi et al., 2021) as follows

$$\begin{aligned}
&\tilde{k}_{uu}(x) = (E_{min} + x^{p_{uu}}(E_0 - E_{min}))\,\tilde{k}_{uu}\\
&\tilde{k}_{u\phi}(x,P) = (e_{min} + x^{p_{u\phi}}(e_0 - e_{min}))(2P-1)^{p_P}\tilde{k}_{u\phi}\\
&\tilde{k}_{\phi\phi}(x) = (\varepsilon_{min} + x^{p_{\phi\phi}}(\varepsilon_0 - \varepsilon_{min}))\tilde{k}_{\phi\phi}\\
&\tilde{m}(x) = x\tilde{m}
\end{aligned} \tag{38}$$

where E_{min}, e_{min} and ε_{min} are small numbers to define the minimum values for stiffness, coupling and dielectric matrices while E_0, e_0 and ε_0 are equal to one to define the maximum values of the respective matrices. x is the density variable of each element which varies between zero and one. P is the polarization variable which also varies between zero and one and determines the direction of polarization. p_{uu}, $p_{u\phi}$, $p_{\phi\phi}$ and p_P are penalization factors for the stiffness, coupling, dielectric matrices and polarization value respectively. The interpolation scheme for passive material can be obtained from interpolation scheme (38) knowing that $\tilde{k}_{u\phi}(x,P) = 0$ and $\tilde{k}_{\phi\phi}(x) = 0$.

The rest of this section regarding the implementation of the SIMP approach will be divided into two parts: (1) actuation and (2) energy harvesting.

3.2 Actuation

The optimization problem for actuation starts by defining the objective function. A simple objective function for a compliant mechanism is to maximize the deflection in a desired direction while considering a volume constraint on the total density variables

$$\begin{aligned}
&minimize \quad J_{act} = -L^T\tilde{U}\\
&Subject\ to \quad V(x) = \sum_{i=1}^{NE} x_i v_i \leq V\\
&0 < x_i \leq 1\\
&0 \leq P_i \leq 1
\end{aligned} \tag{39}$$

where L is a Boolean vector with some values of one that correspond to the output displacement node and zero otherwise. V is the target volume as a fraction of the overall volume of the design domain while v_i is the volume of each element and NE is the total number of elements while i is the number of each element. It should be noted that there are various objective functions for compliant mechanisms which are reviewed by Zhu et al. (2020). We chose a simple objective function with a modeled spring to simulate the stiffness of the target object. The sensitivity analysis regarding the defined objective function with the help of adjoint method can be expressed as

$$\begin{aligned}
&\frac{\partial J}{\partial x_i} =\\
&\frac{\partial}{\partial x_i}\left(-L^T\tilde{U} + \lambda_i^T\left(\tilde{K}_{uu}\tilde{U} + \tilde{K}_{u\phi_{fn}}\tilde{\Phi}_{fn} - \tilde{F} + \tilde{K}_{u\phi_{en}}\tilde{\Phi}_{en}\right)\right.\\
&\left.+\mu_i^T\left(\tilde{K}_{\phi_{fn}u}\tilde{U} - \gamma\tilde{K}_{\phi_{fn}\phi_{fn}}\tilde{\Phi}_{fn} - \gamma\tilde{K}_{\phi_{fn}\phi_{en}}\tilde{\Phi}_{en}\right)\right)
\end{aligned} \tag{40}$$

where λ_i and μ_i are the elemental format of the global adjoint vectors Λ and Υ. The global adjoint vectors are computed by solving of the following adjoint equation,

$$\begin{bmatrix} \tilde{K}_{uu} & \tilde{K}_{u\phi_{fn}} \\ \tilde{K}_{\phi_{fn}u} & -\gamma\tilde{K}_{\phi_{fn}\phi_{fn}} \end{bmatrix}\begin{bmatrix} \Lambda \\ \Upsilon \end{bmatrix} = \begin{bmatrix} L \\ 0 \end{bmatrix} \tag{41}$$

It should be noted that the sensitivity equation (40) includes elemental stiffness matrices which can be different for the passive and active materials. Thus, the sensitivity can be written as

$$\begin{aligned}
&\frac{\partial J}{\partial x_i} = \lambda_i^T\frac{\partial\tilde{k}_{uu}}{\partial x_i}\tilde{u_i} + \lambda_i^T\frac{\partial\tilde{k}_{u\phi}}{\partial x_i}\tilde{\phi}_i\\
&+\mu_i^T\frac{\partial\tilde{k}_{\phi u}}{\partial x_i}\tilde{u}_i - \gamma\mu_i^T\frac{\partial\tilde{k}_{\phi\phi}}{\partial x_i}\tilde{\phi}_i
\end{aligned} \tag{42}$$

where the corresponding elemental matrices for active and passive elements should be used and in this case

$k_{u\phi\{p\}} = 0$. We will not repeat this fact for the rest of the sensitivity analysis. The sensitivity analysis with respect to polarization P is

$$\frac{\partial J}{\partial P_i} = \lambda_i^T \frac{\partial \tilde{k}_{u\phi}}{\partial p_i} \tilde{\phi}_i + \mu_i^T \frac{\partial \tilde{k}_{\phi u}}{\partial p_i} \tilde{u}_i \qquad (43)$$

where λ_i and μ_i are already calculated in the adjoint equation (40).

Based on equations (40) and (43), the derivative of piezoelectric stiffness and coupling matrices with respect to design variables are required, which can be derived with the help of equation (38) as

$$\frac{\partial \tilde{k}_{uu}}{\partial x_i} = p_{uu}(E_0 - E_{min}) x_i^{p_{uu}-1} \tilde{k}_{uu} \qquad (44)$$

$$\frac{\partial \tilde{k}_{u\phi}}{\partial x_i} = p_{u\phi}(e_0 - e_{min}) x_i^{p_{u\phi}-1} (2P_i - 1)^{p_P} \tilde{k}_{u\phi} \qquad (45)$$

$$\frac{\partial \tilde{k}_{u\phi}}{\partial P_i} = 2p_P(e_0 - e_{min})(2P_i - 1)^{p_P - 1} x_i^{p_{u\phi}} \tilde{k}_{u\phi} \qquad (46)$$

After obtaining the sensitivity analysis, we can use gradient-based optimization solvers to update the optimization variables. This will be discussed after formulating the optimization problem for the energy harvesting case.

3.3 Energy Harvesting

The objective function for energy harvesting can be considered as electromechanical efficiency (electrical energy/mechanical energy) (Zheng et al., 2009; Noh and Yoon, 2012). However, to have a more mechanically stable topology and more smooth convergence to final topology, the minimization of the weighted sum of the mechanical and electrical energies is considered as objective function (Homayouni-Amlashi et al., 2019)

$$\begin{aligned} &minimize \quad J_{EH} = w_j \mathit{\Pi}^S - (1 - w_j)\mathit{\Pi}^E \\ &Subject\,to \quad V(x) = \sum_{i=1}^{NE} x_i v_i \le V \\ &0 < x_i \le 1 \\ &0 \le P_i \le 1 \end{aligned} \qquad (47)$$

In which, w_j is the weighing factor that has a value between zero and one and can be determined by a trial-error procedure. $\mathit{\Pi}^S$ and $\mathit{\Pi}^E$ are the mechanical and electrical energies defined as

$$\begin{aligned} &\mathit{\Pi}^S = (\frac{1}{2}) \tilde{U}^T \left[\tilde{K}_{uu} - \tilde{M}\tilde{\mathit{\Omega}}^2\right] \tilde{U}, \\ &\mathit{\Pi}^E = (\frac{1}{2}) \gamma \tilde{\mathit{\Phi}}^T \tilde{K}_{\phi\phi} \tilde{\mathit{\Phi}} \end{aligned} \qquad (48)$$

Based on the defined objective function for the energy harvesting mode, the sensitivity of the objective function with respect to density can be calculated as (Zheng et al., 2009; Homayouni-Amlashi et al., 2021),

$$\begin{aligned} &\frac{\partial \mathit{\Pi}^S}{\partial x_i} = (\frac{1}{2}\tilde{u}_i^T + \lambda_{1,i}^T) \frac{\partial(\tilde{k}_{uu} - \tilde{m}\tilde{\mathit{\Omega}}^2)}{\partial x_i} \tilde{u}_i + \\ &\lambda_{1,i}^T \frac{\partial \tilde{k}_{u\phi}}{\partial x_i} \tilde{\phi}_i + \mu_{1,i}^T \frac{\partial \tilde{k}_{\phi u}}{\partial x_i} \tilde{u}_i - \mu_{1,i}^T \frac{\gamma \partial \tilde{k}_{\phi\phi}}{\partial x_i} \tilde{\phi}_i \end{aligned} \qquad (49)$$

$$\begin{aligned} &\frac{\partial \mathit{\Pi}^E}{\partial x_i} = \frac{1}{2} \tilde{\phi}_i^T \frac{\gamma \partial \tilde{k}_{\phi\phi}}{\partial x_i} \tilde{\phi}_i - \mu_{2,i}^T \frac{\gamma \partial \tilde{k}_{\phi\phi}}{\partial x_i} \tilde{\phi}_i + \\ &\lambda_{2,i}^T \frac{\partial(\tilde{k}_{uu} - \tilde{m}\tilde{\mathit{\Omega}}^2)}{\partial x_i} u_i + \lambda_{2,i}^T \frac{\partial \tilde{k}_{u\phi}}{\partial x_i} \tilde{\phi}_i + \mu_{2,i}^T \frac{\partial \tilde{k}_{\phi u}}{\partial x_i} \tilde{u}_i \end{aligned} \qquad (50)$$

in which μ and λ are the elemental adjoint vectors which are calculated by the following global coupled system

$$\begin{aligned} &[S_{EH}] \begin{bmatrix} \mathit{\Lambda}_1 \\ \mathit{\Upsilon}_{1,fn} \\ \mathit{\Upsilon}_{1,en} \end{bmatrix} = \begin{bmatrix} -\left(\tilde{K}_{uu} - \tilde{M}\tilde{\mathit{\Omega}}^2\right)\tilde{U} \\ 0 \\ 0 \end{bmatrix} \\ &[S_{EH}] \begin{bmatrix} \mathit{\Lambda}_2 \\ \mathit{\Upsilon}_{2,fn} \\ \mathit{\Upsilon}_{2,en} \end{bmatrix} = \begin{bmatrix} 0 \\ -\gamma \tilde{K}_{\phi_{fn}\phi_{fn}} \tilde{\mathit{\Phi}}_{fn} - \gamma \tilde{K}_{\phi_{fn}\phi_{en}} B V_p \\ -\gamma B^T \tilde{K}_{\phi_{en}\phi_{fn}} \tilde{\mathit{\Phi}}_{fn} - \gamma B^T \tilde{K}_{\phi_{en}\phi_{en}} B V_p \end{bmatrix} \end{aligned} \qquad (51)$$

where $\mathit{\Lambda}$ and $\mathit{\Upsilon}$, are the global adjoint vectors and $[S_{EH}]$ is already defined in equation (36). The sensitivities with respect to polarization (P) are calculated as well (Homayouni-Amlashi et al., 2020, 2021)

$$\begin{aligned} &\frac{\partial \mathit{\Pi}^S}{\partial P_i} = \lambda_{1,i}^T \frac{\partial \tilde{k}_{u\phi}}{\partial P_i} \tilde{\phi}_i + \mu_{1,i}^T \frac{\partial \tilde{k}_{\phi u}}{\partial P_i} \tilde{u}_i \\ &\frac{\partial \mathit{\Pi}^E}{\partial P_i} = \lambda_{2,i}^T \frac{\partial \tilde{k}_{u\phi}}{\partial P_i} \tilde{\phi}_i + \mu_{2,i}^T \frac{\partial \tilde{k}_{\phi u}}{\partial P_i} \tilde{u}_i \end{aligned} \qquad (52)$$

Based on sensitivity equations (50) and (60), the derivatives of all piezoelectric matrices with respect to the design variables are required. The derivative of stiffness and coupling matrices are found in equations (45) and (46). Here, the derivative of the dielectric matrix and mass matrix is also required which are

$$\frac{\partial \tilde{k}_{\phi\phi}}{\partial x_i} = p_{\phi\phi}(\varepsilon_0 - \varepsilon_{min})x_i^{p_{\phi\phi}-1}\tilde{k}_{\phi\phi}$$
$$\frac{\partial \tilde{m}}{\partial x_i} = \tilde{m}_i \tag{53}$$

In addition to the derivative of piezoelectric matrices with respect to density, derivation of the piezoelectric coupling matrix with respect to polarization variable is also required

$$\frac{\partial \tilde{k}_{u\phi}}{\partial P_i} = 2p_P(2P_i - 1)^{p_P-1}x_i^{p_{u\phi}}\tilde{k}_{u\phi} \tag{54}$$

We remind here once again that in the sensitivity analysis, the corresponding elemental stiffness matrices for the passive and active elements should be used. This will be discussed in section 4 as well. After the calculation of sensitivities, the optimization variables can be updated in each iteration of optimization with the help of the optimization algorithm which is the subject of the next section.

3.4 Optimization algorithm

There are different algorithms for updating the optimization variables in each iteration of topology optimization. The most used ones are the optimality criteria (OC) (Venkayya, 1989; Bendsøe and Sigmund, 2003) or the Method of Moving Asymptotes (MMA) developed by Svanberg (1987, 2007). Aforementioned algorithms are used in the previously published codes by authors (Homayouni-Amlashi et al., 2021). In this paper, we used the routine recently developed by Ferrari et al. (2021) which is a sequential approximation approach, using monotonic MMA-like approximations and an OC-like scheme. By using this routine, the implementation codes in the appendix can be run solely without calling any external codes.

After providing the theoretical background for the application of TO to piezoelectric actuators and energy harvesters in the previous sections, it is now possible to present the implementation MATLAB codes in the next section.

4 MATLAB Implementation Codes

In this section, the MATLAB implementation codes which are provided in the appendix will be explained in detail part by part. Two main codes are provided in the appendix: one for actuation which is called `3DTOPIEZO_ACTUATION` and the other for the energy harvesting which is called `3DTOPIEZO_ENERGY-HARVESTING`. The explanations of each part of the code will be followed by the corresponding lines of code with the attached numbers where it is necessary. In this case, the color of the number for each line shows that the line belongs to actuation code or energy harvesting code. These two main codes need five add-on functions which are provided in the appendix as well. It should be noted that by adding these add-on functions to the main codes, these codes can be run without calling any other external functions or codes. These two main codes and the add-on functions are provided as supplementary materials for this paper.

4.1 Actuation

The default actuation code, which is provided in the appendix, is developed for a bi-morph bending actuator in Fig. 10. Other examples and case studies are provided as separate codes as supplementary materials to this paper.

4.1.1 General Definition

The first part of the code is `GENERAL DEFINITIONS`. The geometry of the active and passive domains (piezo and non-piezo) are defined separately. `La, Wa` are the length and width of all the piezoelectric layers while `Ha` is the thickness of the element-wise layer of piezoelectric layer. `Lp, Wp, Hp` are the geometry dimensions of passive material and `Hp` is the element-wise thickness. As it was illustrated in Fig. 2, we define the height of one element in the direction of thickness while the length and width will be defined for whole of the layer. The length and width of the active and passive domains should be equal for now and just the height (thickness) can be defined differently. The resolution of the mesh for the total design domain is defined by `nelx, nelz, nely`. `penalKuu, penalKup, penalPol`, are the penalization factors which defines the $p_{uu}, p_{u\phi}, p_{\phi\phi}$ respectively. For choosing these penalization factors, the same criteria which were provided by Kim et al. (2010) and mentioned in (Homayouni-Amlashi et al., 2021) are used here. `EL_T` specifies the element type which accepts two values of `1` and `2` to choose the trilinear and quadratic elements respectively.

There are other parameters that are common between the Matlab codes i.e. `volfrac, Max_loop` which specify the volume fraction and maximum iteration loops respectively. It has been explained in the previous educational paper by Homayouni-Amlashi et al. (2021) that two stopping criteria can be defined for the codes. The classical criteria is the negligible density change between the two last successive iterations and the other criteria is the maximum number of iterations. However, we experienced that the density change will not stop the optimization generally or it needs high number of iterations while there are no significant changes in the objective function's value. For this reason, the definition of the maximum allowed iteration loop is essential.

`pol_dir` is a new variable in comparison to previous published codes (Homayouni-Amlashi et al., 2021). This variable defines the direction of the polarization which by default accepts the values of `'x','y','z'`. As explained before, in the data sheets for the piezoelectric materials, the $\vec{z}$ axis is usually the polarization direction.

`rmin, eta, beta` are parameters that are related to the filtering that stand for filter radius, threshold and sharpness factor. `ftBC` is the boundary condition for the built-in MATLAB code `imfilter` which accepts `'N'` for zero-Neumann or `'D'` for zero-Dirichlet. The explanation about these parameters and their values can be found in the works by Ferrari and Sigmund (2020) and Wang et al. (2011).

In this paper, the continuation scheme will be applied to the penalties and sharpness factor. To do so, `penalCnt, betaCnt` are defined similar to what have been defined by Ferrari and Sigmund (2020). These parameters accept four values as `[istart, maxPar, isteps, deltaPar]`, which means the continuation starts at iteration = `istart` and will be increased by `deltaPar` in each `isteps` and reaching to maximum value `maxPar`.

Finally, there are parameters related to actuation mode as a form of compliant mechanism. `Dir` defines the desired direction of movement for the actuation and `Ks` is the stiffness of the spring attached at the desired points which simulates the reaction force by the workpiece.

4.1.2 Material Properties

In this part of the code, the material properties of the piezoelectric and passive materials are defined. The properties include density, piezoelectric tensor matrices, Poisson ratio and Young's modulus of elasticity for passive material. The default material for PZT is considered to be PZT 4. Changing the material properties for the piezoelectric or passive material affects the final results.

4.1.3 Prepare finite element analysis

In this part of the code, first, the new piezoelectric tensor matrices will be obtained by the rotation of coordinate system through calling the add-on function `Matrix_Rotation` as follows,

```
[C_p_1,e_1,Ep_1] = Matrix_Rotation(C_p,e,
    Ep,pol_dir); % New tensor matrices
    after Rotation of coordinate system
```

where `C_p, e, Ep` are the piezoelectric mechanical, coupling and permittivity matrices. Here, it is assumed that the passive material is isotropic. As such, the rotation of the coordinate system is not necessary for passive material. However, if the passive material is non-isotropic then its matrices should be rotated as well.

By updating the tensor matrices, the finite element matrices will be calculated by calling the add-on function `FEM`.

```
[kuu,kup,kpp,~,ndofPZT,EL_NN,TOPNODS,
    BOTNODS,FRNODS,BAKNODS,LEFNODS,RTNODS]
    = FEM(La,Wa,Ha,nelz,nelx,nely,C_p_1,e_1
    ,Ep_1,0,EL_T); % Piezoelectric
    elemental matrices
[ks,~,~,~,~,~,~,~,~,~,~,~] = FEM(Lp,Wp,Hp,
    max(1,nelz-2),nelx,nely,C_s,zeros(3,6)
    ,zeros(3,3),0,EL_T); % Passive material
     elemental matrices
```

This function produces the finite element matrices based on the geometry of piezoelectric and passive material and based on the resolution of the defined mesh. Other outputs are `ndofPZT` which is the total number of nodes and total number of electrical DOFs, `EL_NN` is the total number of nodes for one element which is 8 for linear elements and 20 for quadratic elements. `TOPNODS, BOTNODS, FRNODS, BAKNODS, LEFNODS, RTNODS` are the internal node numbers for the 8 faces of an element based on Fig. 6. For example, for linear elements, based on Fig . 3-(c), the node numbers for the top face of the element which is specified by `TOPNODS` are [3,4,7,8].

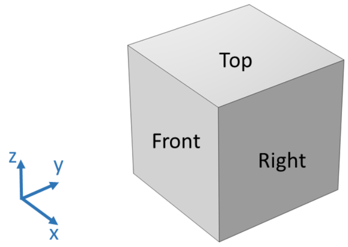

Fig. 6 Element face indexing.

After calculation of the elemental matrices, normalization is applied based on equation (26) similar to the previously published codes (Homayouni-Amlashi et al., 2021). The normalization factors are saved for the recreation of matrices and obtaining the real response values.

```matlab
k0 = max(abs(kuu(:)));beta0 = max(kpp(:)); alpha0 = max(kup(:)); % Normalization Factors
kuu = kuu/k0;ks = ks/k0;kup = kup/alpha0; kpp = kpp/beta0;gamma = (k0*beta0)/(alpha0^2); % Application of normalization
```

To speed up the assembly procedure and decrease memory usage, the procedure which, was recently introduced by Ferrari and Sigmund (2020), is used to build the lower triangular stiffness matrices. To do so, first, the vectors of lower triangular stiffness matrices are built excluding the coupling matrix which is not a square matrix. These vectors will be used later for the assembly procedure to build the global matrices.

```matlab
kuu_LT = kuu(tril(true(size(kuu)))); % Vector of lower triangular matrix
kpp_LT = kpp(tril(true(size(kpp)))); % Vector of lower triangular element of piezoelectric dielectric stifness matrix
ks_LT = ks(tril(true(size(ks)))); % Vector of lower triangular matrix
```

`ndof, nele` are the total number of mechanical DOFs and total number of elements respectively. The indexing matrix (`ElNum`) is defined in this paper to index all the elements in the design domain based on the numbering format which has been illustrated in Fig. 3.

This 3-dimensional indexing matrix helps specify the elements in the design domain using three inputs as follows,

$$ElNum\,(z\ elements, x\ elements, y\ elements) \tag{55}$$

The first input specifies the $\vec{z}$ coordinate of the element, and the second and third inputs specify the $\vec{x}$ and $\vec{y}$ coordinate of the element in the design domain. For example, for the coarse mesh in Fig. 3, the elements of the `ElNum` can be obtained by,

$$ElNum\,(:,:,1) = \begin{bmatrix} 1\ 3\ 5\ 7 \\ 2\ 4\ 6\ 8 \end{bmatrix} \tag{56}$$

$$ElNum\,(:,:,2) = \begin{bmatrix} 9\ \ 11\ 13\ 15 \\ 10\ 12\ 14\ 16 \end{bmatrix} \tag{57}$$

These matrices correspond to the element numbering in Fig. 3. For example, to specify the element number 6 in the design domain, it is enough to write `ElNum(2,3,1)` which produces 6 as output. Since element 6 lies in the second row in the $\vec{z}$ direction, the third row in the $\vec{x}$ direction and the first row in the $\vec{y}$ direction. The indexing matrix `ElNum` will be used many times later to facilitate the specifications of nodes and DOFs in different parts of the codes.

The mechanical connectivity matrix named as `edofMat` is for indexing the mechanical DOFs and it is defined for trilinear elements in the code published by Liu and Tovar (2014) while it was originally proposed by Andreassen et al. (2011) for 2D quadrilateral elements. In this paper, we proposed our flexible mechanical connectivity matrix. Indeed, depending on the element node numbers (`EL_NN`) the procedure to build the mechanical connectivity matrix is flexible to switch from trilinear to quadratic elements. Each row of this matrix belongs to one element and the columns show the mechanical DOFs related to that element,

$$edofMat = \begin{array}{r} element\ 1 \longrightarrow \\ element\ 2 \longrightarrow \\ \vdots \\ element\ N_e \longrightarrow \end{array} \overbrace{\begin{bmatrix} [] & [] & \cdots & [] \\ [] & [] & \cdots & [] \\ \vdots & \vdots & \ddots & \vdots \\ [] & [] & \cdots & [] \end{bmatrix}}^{Mechnical\ DOFs} \tag{58}$$

For piezoelectric materials, in addition to mechanical DOFs, we have electrical DOFs. As it has been shown in panel (b) of Fig. 3, each node has one electrical DOF and the numbering of electrical DOFs and the numbering of the nodes are the same. Therefore, the electrical connectivity matrix known as `edofMatPZT` is a matrix in which each row belongs to one element while the columns are the node numbers related to that element.

$$edofMatPZT = \begin{array}{r} element\ 1 \longrightarrow \\ element\ 2 \longrightarrow \\ \vdots \\ element\ N_e \longrightarrow \end{array} \overbrace{\begin{bmatrix} [] & [] & \cdots & [] \\ [] & [] & \cdots & [] \\ \vdots & \vdots & \ddots & \vdots \\ [] & [] & \cdots & [] \end{bmatrix}}^{Node\ Numbers} \tag{59}$$

To assemble the global stiffness matrix for passive material Ferrari and Sigmund (2020) proposed an assembly indexing matrix `Iar` which takes advantage of the symmetry of the stiffness matrix. Here, by introducing `EL_NN`, we make it flexible for considering trilinear or quadratic elements.

```matlab
[ sI, sII ] = deal( [ ] );
for j = 1 : 3*EL_NN
sI = cat( 2, sI, j : 3*EL_NN );
sII = cat( 2, sII, repmat( j, 1, 3*EL_NN - j + 1 ) );
end
```

```
[ iK , jK ] = deal( edofMat( :,  sI )',
     edofMat( :, sII )' );
Iar = sort( [ iK( : ), jK( : ) ], 2, '
     descend' ); clear iK jK % Assembly
     indexing (stiffness matrix)
```

To adapt the same procedure for building the piezoelectric coupling matrix, the related indexing matrix should be built by considering the fact that we have electrical DOFs equal to number of nodes while mechanical DOFs are three times more. As such the routine to build the assembly indexing matrix for piezoelectric coupling (`Iar_up`) matrix is as follows,

```
[ sI, sII ] = deal( [ ] );
for j = 1 : EL_NN
sI = cat( 2, sI, 1 : 3*EL_NN );
sII = cat( 2, sII, repmat( j, 1, 3*EL_NN  )
     );
end
[ iKup , jKup ] = deal( edofMat( :,  sI )
     ', edofMatPZT( :, sII )' );
Iar_up = [ iKup( : ), jKup( : ) ] ; clear
     iKup jKup; % Assembly indexing for
     piezoelectric coupling matrix
```

Indexing matrix for piezoelectric permittivity matrix is defined by `Iar_p` considering electrical DOFs,

```
[ sI, sII ] = deal( [ ] );
for j = 1:EL_NN
sI = cat(2,sI,j:EL_NN);
sII = cat(2,sII,repmat(j,1,EL_NN-j+1));
end
[iKp,jKp] = deal(edofMatPZT(:,sI)',
     edofMatPZT(:,sII)');
Iar_p = sort([iKp(:),jKp(:)],2,'descend');
      clear iKp jKp % Assembly indexing for
     piezoelectric dielectric stiffness
     matrix
```

4.1.4 New indexing method for nodes and DOFs

In the provided codes of this paper, an efficient and compact method for indexing the elements, nodes and mechanical DOFs is proposed which facilitates the definition of boundary conditions, applied forces, electrodes, etc. This indexing method uses the indexing matrices `ElNum`, `edofMat` and `edofMatPZT`. To better understand this indexing method, we have to inspect once again the `edofMat`, `edofMatPZT`. Let's consider the element number 12 in the coarse discretization of Fig. 3 with trilinear elements. The `edofMatPZT(12,:)` will show all the columns in the row number 12 of the `edofMatPZT` as follows

$$\begin{array}{rl} & \begin{matrix}1&2&3&4&5&6&7&8\end{matrix} \leftarrow \textit{Element Internal Numbers}\\ edofMatPZT\,(12,:) = & [21\ 24\ 23\ 20\ 36\ 39\ 38\ 35] \leftarrow \textit{Element External Numbers}\end{array}$$

The numbers of nodes are placed in the matrix `edofMatPZT(12,:)` based on the elemental internal numbers. Here, we remind again that the elemental internal numbering and external numbering are different based on Fig. 3, panels (b) and (c). For example, based on panel (c) of Fig. 3, The internal node number 1 corresponds to node number 21 based on the external numbering of the nodes in panel (b). This means that the external node numbers are placed in the columns based on the sequence of internal numbers. The same sequence is true for placement of mechanical DOFs in the columns of `edofMat(12,:)`. To obtain the sequence for mechanical DOFs, it is enough to multiply the node numbers in `edofMatPZT` by 3 and consider the last two previous sequences. In this regard, the 3 mechanical DOFs corresponding to node number 21 are 61, 62 and 63. The same goes for other node numbers.

$$\begin{array}{rl} & \{1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\,11\,12\ 13\ 14\ 15\ 16\ 17\ 18\ 19\ 20\ 21\ 22\ 23\ 24\} \leftarrow \textit{Element Internal Numbers}\\ edofMat\,(12,:) = & [61\ 62\ 63\ 70\ 71\ 72\ 67\ 68\ 69\ 58\ 59\ 60\ 106\ 107\ 108\ 115\ 116\ 117\ 112\ 113\ 114\ 103\ 104\ 106] \leftarrow \textit{Element External Numbers}\end{array}$$

The aforementioned examples were for trilinear elements. For quadratic elements, the node numbers continue until 20 and the mechanical DOFs correspond to those in Fig. 4-(e). After understanding how the indexing matrices `edofMat`, `edofMatPZT` are built, we can introduce our indexing method with the help of elemental indexing matrix `ElNum`. We follow a sequence of steps to determine the electrical and mechanical DOFs: 1) specifying the elements, 2) specifying the nodes and 3) specifying the mechanical DOFs. This will be explained in the next sections and readers are advised to check the indexing method in each part on the coarse discretization of Figs. 3 and 4 to better understand the method.

4.1.5 Active and passive domains

In this section, the goal is to define the active and passive domains which specify the piezoelectric and non-piezoelectric domains respectively. To do so, we just need to specify the elements that belong to each of these two domains using the `ElNum`.

```
Passive_el = ElNum([] ,:,:); Passive_el = Passive_el(:); % Definition of passive elements
Active_el = setdiff(1:nele,Passive_el); Active_el = Active_el(:);% Definition of active elements
```

In the lines above, we determined the elements that belong to passive domain using the (`Passive_el`). Then, the rest of elements will belong to the active domains.

4.1.6 Boundary condition

In this section of the code, the goal is to define the mechanical boundary condition using the proposed indexing method. Let's suppose that the coarse design domain of Fig. 3 is clamped on the left side and we want to specify all the fixed mechanical DOFs attached to the clamped side. First, we have to specify the desired elements.

```
DE = ElNum(:,1,:); DE=DE(:); % Desired elements for left clamped side
```

`ElNum(:,1,:)` specifies all the elements in the $\vec{z}$ direction and all the elements in the $\vec{y}$ direction for the first row of elements in the $\vec{x}$ direction. For the design domain in Fig. 3, it will produce element numbers [1,2,9,10] which are the elements that are attached to the left clamped side. The next step toward the definition of mechanical DOFs is the specification of the nodes. We previously noted that the placement of node numbers and the corresponding mechanical DOFs in the `edofMatPZT`, `edofMat` are based on the internal node numbering sequence. Therefore, we specify the internal node numbers that belong to the left side of the element,

```
DNN = LEFNODS; % Desired node numbers (elemental left nodes)
```

Now, to specify the mechanical DOFs, we use the `edofMat` matrix,

```
fixeddof = edofMat(DE,[3*DNN,3*DNN-1,3*DNN-2]); fixeddof = unique(fixeddof(:)); % Fix mechanical DOFs
```

Indeed, from the rows of `edofMat`, we chose those rows that we previously defined as the desired elements which are attached to the clamped side. From the columns, we multiply the node numbers by three and consider the two previous consecutive numbers. This will successfully determine the fixed mechanical DOFs. By having the fixed mechanical DOFs, the free mechanical DOFs can be obtained by excluding the fixed DOFs from all DOFs (`NDOF`).

4.1.7 Definition of electrodes

In this section, we have to define the equipotential nodes (en) and free nodes as it was illustrated in Fig. 5. To do so, we consider a two-layer (bi-morph) piezoelectric plate of Fig. 10-(a). In this case, we have 3 electrodes in total that we recognize as top, bottom and middle electrodes.

To apply the equipotential boundary conditions, the nodes which are connected to these electrodes should be specified. In this regard, the first step is to specify the elements connected to each electrode

```
DE = ElNum(1,:,:); DE = DE(:); % Desired elements for top electrode
DE2 = ElNum(nelz,:,:); DE2 = DE2(:); % Desired elements for bottom electrode
DE3 = ElNum(ceil(nelz/2),:,:); DE3 = DE3(:); % Desired elements for Middle electrode
```

After specifying the element, the corresponding nodes that are connected to each electrode should be specified,

```
TE = edofMatPZT(DE,TOPNODS); TE = unique(TE(:)); % Top electrode
BE = edofMatPZT(DE2,BOTNODS); BE = unique(BE(:)); % Bottom electrode
ME = edofMatPZT(DE3,BOTNODS); ME = unique(ME(:)); % Mid electrode
```

By `edofMatPZT(DE,TOPNODS)`, we chose the elements that are attached to top electrodes and in those element we chose the nodes on the top surface of the element by `edofMatPZT(DE,TOPNODS)` and for the bottom elements, the internal nodes of the bottom face are chosen by `BOTNODS`. The same procedure is done for the middle and bottom electrode. However, in addition to nodes that are connected to electrodes, there are nodes that are not connected to any electrodes and there are nodes which belong to passive materials as they are illustrated in Fig. 5.

```
en = [TE;BE;ME]; en = unique(en(:)); % Equipotential nodes
pn = edofMatPZT(Passive_el,:); pn = unique(setdiff(pn(:),en(:))); % Nodes of passive elements
fn = setdiff(1:ndofPZT,[en;pn]); fn = unique(fn(:)); % FreeNodes
```

(`en`) contains the equipotential nodes that are connected to electrodes, (`pn`) contains the nodes belongs to passive material which will have zero potential and (`fn`) are the free nodes in the piezoelectric material not connecting to any electrodes.

In the actuation example of Fig. 10-(a), we apply the voltage to the top and bottom electrodes and impose the ground condition on the middle electrode,

```
Up([TE;BE],1) = 1; % Actuation voltage
```

where (`Up`) is the vector of the electric potentials.

4.1.8 Output displacement definition

This section explains how to define the desired displacement in the specific point of the design. Based on the default actuation code which is provided in the appendix, the goal is to have a deflection in the $\vec{z}$ direction at the tip of the piezoelectric bi-morph in the middle as shown in Fig. 10-(a). In this case, we have to specify the desired mechanical DOFs. For choosing the elements, nodes and DOFs, following lines of codes are written,

```
DE = ElNum(ceil(nelz/2),nelx,ceil(nely/2))
    ; % Desired element
DNN = 6;% Desired node numbers
DMDOF = edofMat(DE,3*DNN-Dir); DMDOF=
    unique(sort(DMDOF(:))); % Desired
    mechanical degree of freedom
L = sparse(ndof,1); L(DMDOF,1) = -1;
```

This means that we chose the middle element in the $\vec{y}$ direction at the tip of the beam in the $\vec{x}$ direction and at the center with respect to the $\vec{z}$ direction. To choose the desired node, we chose the internal node number 6 based on the element's internal node numbering Fig. 3-(c).

(`Dir`) is the direction of the desired displacement which is defined in the part (`General definitions`) that can accept the values of 0, 1 and 2 corresponding the displacement in the $\vec{x}$, $\vec{y}$ and $\vec{z}$ directions. `DMDOF` finally gives us the desired mechanical DOFs for displacement and we can define the vector L in the objective function (39).

4.1.9 Solid and void domains

Similar to classical codes for compliance problems (Andreassen et al., 2011; Sigmund, 2001a; Liu and Tovar, 2014), there are areas inside the design domain that can be considered solid or void domains. The elements in these areas will not be updated in the optimization iterations. To specify the elements which will be updated by the optimization algorithm, we defined `NVS` as a vector of all non-void and non-solid elements.

4.1.10 Filter Initialization

The filtering parts of the codes are using the same lines of codes developed by Ferrari and Sigmund (2020) for compliance problems and readers are referred to the work by Wang et al. (2011) to deeply understand the theoretical concepts.

4.1.11 Initialize iteration

Before starting the optimization iterations, some parameters are defined in this part. For example, initial values for the densities (`x`) and polarization (`pol`).

Since we are using a continuation scheme, the penalty for stiffness matrix (`penalKuu`) will increase during the optimization. On the other hand, to respect the criteria for the penalization factors which have been mentioned in the previously published codes for piezoelectrics in 2D (Homayouni-Amlashi et al., 2021), we defined the variables `penalratio_up`, `penalratio_pp` to keep the ratio between different penalization factors constant during the optimization,

```
penalratio_up = penalKup/penalKuu;
    penalratio_pp = penalKpp/penalKuu;  %
    Penalty ratios for continuation scheme
```

4.1.12 Optimization iterations

This is the main part of the code where the optimization variables converge to their optimized values and the optimized topology will be obtained in a sequence of iterations. This part starts with the computation of the physical density field (and eta if projection is chosen i.e. ft=2 or 3) which is written by Ferrari and Sigmund (2020). The next part is the finite element analysis which builds the global stiffness matrices and obtains the response of the system. To build the global stiffness matrices, the following lines of codes are written,

```
sK = ones(length(kuu_LT( : )),1).*(Emin+
    xPhys(:)'.^penalKuu*(E0-Emin));
sK(:,Active_el) = kuu_LT( : ).* sK(:,
    Active_el);
sK(:,Passive_el) = ks_LT( : ).* sK(:,
    Passive_el);
```

The matrix `sK` is firstly defined in the codes published by Andreassen et al. (2011) in which the number of rows is equal to the length of the vectorized elemental stiffness matrix and the number of columns is equal to the number of total elements in the design domain. This matrix applies the material interpolation function which is introduced in equation (38) using the updated density (`xPhys`) and polarization (`pol`). However, a modification is made to this matrix to include the active and passive material. Those columns of matrix `sK` which belong to the active domain are multiplied by the piezoelectric elemental stiffness matrix (`kuu_LT`) and those which belong to the passive material are multiplied by the passive elemental stiffness matrix (`ks_LT`).

The same strategy is used to build the `sKup` and `sKpp` which are used to build the global coupling and permittivity matrices.

```
sKup = ones(length(kup(:)),1).*(eMin+xPhys(:)'.^penalKup*(e0-eMin).*((2*pol(:)-1)'.^penalPol));
sKup(:,Active_el) = kup(:).* sKup(:,Active_el);
sKup(:,Passive_el) = 0;
sKpp = ones(length(kpp_LT(:)),1)*(epsMin+xPhys(:)'.^penalKpp*(eps0-epsMin));
sKpp(:,Active_el)= kpp_LT(:).* sKpp(:,Active_el);
sKpp(:,Passive_el)= 0;
```

However, the columns of sKup and sKpp which belong to the passive material are considered to be zero since there is no coupling between the mechanical and electrical part. In the next step, by using the indexing matrices Iar, Iar_up and Iar_p which are explained before, the global matrices are built.

```
Kuu = sparse(Iar( :, 1 ),Iar( :, 2 ),sK(:)); Kuu = Kuu+Kuu'-diag(diag(Kuu)); % Global stifness matrix
for i=1: length(DMDOF);Kuu(DMDOF(i,1),DMDOF(i,1)) = Kuu(DMDOF(i,1),DMDOF(i,1))+Ks; end % Assembling the stifness of the modeled spring
Kup = sparse(Iar_up(:,1),Iar_up(:,2),sKup(:)); % Global piezoelectric coupling matrix
Kpp = sparse(Iar_p(:,1),Iar_p(:,2),sKpp(:)); Kpp = Kpp+Kpp'-diag(diag(Kpp)); % Global piezoelectric permittivity matrix
```

The stiffness of the modeled spring which imitates the stiffness of the workpiece is augmented to the global stiffness matrix in the defined mechanical DOFs using the DMDOF as it was defined in the section "output displacement definition".

After the creation of stiffness matrices, it is possible to obtain the response from solving the system of equations (37),

```
Ktot = [Kuu(freedofs,freedofs),Kup(freedofs,fn);Kup(freedofs,fn)',-gamma*Kpp(fn,fn)];
U = Ktot \ [-Kup(freedofs,en)*Up(en,:);gamma*Kpp(fn,en)*Up(en,:)]; % Response of the system
Uu(freedofs) = U(1:length(freedofs)); Up(fn) = U(length(freedofs)+1:end);
```

The vector U, contains full system response which is separated to mechanical Uu and electrical responses Up.

4.1.13 Sensitivity analysis

After obtaining the response of the system by solving the linear equilibrium equation, it is possible to calculate the objective function and perform the sensitivity analysis. For the actuation, the objective function is calculated based on equation (39). To perform the sensitivity analysis, first, the global adjoint equation (41) should be solved and

```
ADJ =  Ktot \[L(freedofs,1);0*fn];
```

After calculation of the adjoint vector, the elemental sensitivities can be computed

```
DCKuuE(Active_el) = sum((lambda(edofMat(Active_el,:))*kuu).*Uu(edofMat(Active_el,:)),2);
DCKuuE(Passive_el) = sum((lambda(edofMat(Passive_el,:))*ks).*Uu(edofMat(Passive_el,:)),2);
DCKupE(Active_el) = sum((lambda(edofMat(Active_el,:))*kup).*Up(edofMatPZT(Active_el,:)),2);
DCKupE(Passive_el) = 0;
DCKpuE(Active_el) = sum((Uu(edofMat(Active_el,:))*kup).*mu(edofMatPZT(Active_el,:)),2);
DCKpuE(Passive_el) = 0;
DCKppE(Active_el) = -gamma*sum((mu(edofMatPZT(Active_el,:))*kpp).*Up(edofMatPZT(Active_el,:)),2);
DCKppE(Passive_el) = 0;
```

In the calculation of elemental sensitivity analysis from equation (40), the elemental stiffness matrix for the active and passive material should be multiplied accordingly. This is considered in the code by using the edofMat and Active_el, Passive_el. Moreover, the elemental coupling matrix for the passive material is considered to be zero and the respective sensitivity will be zero as well.

At last, the sensitivities with respect to density (dc) and with respect to polarization (dp) are calculated. Then, the built-in function imfilter from MATLAB is used to filter the sensitivities as it was proposed by Ferrari and Sigmund (2020).

```
dc = penalKuu*(E0-Emin)*xPhys.^(penalKuu-1).*DCKuu+ penalKup*(E0-Emin)*((2*pol-1).^(penalPol)).*xPhys.^(penalKup-1).*DCKup+penalKpp*(eps0-epsMin)*xPhys.^(penalKpp-1).*DCKpp; % Sensitivity with respect to x
dp = 2*penalPol*((2*pol-1).^(penalPol-1)).*xPhys.^(penalKup).*DCKup; % Sensitivity with respect to p
```

4.1.14 Updating optimization variables

To update the optimization variables using the sensitivity analysis, the ocUpdate routine written and developed by Ferrari et al. (2021) is employed. We employe this updating routine since it lets the provided codes be compiled solely and independently from calling external codes. In the author's previously published codes (Homayouni-Amlashi et al., 2021), the MMA algorithm was necessary to compile the energy harvesting code and users had to request the author of the MMA paper (Svanberg, 1987) to obtain the Matlab implementation code. This problem is solved in the current paper by using the updating routine from the code published by Ferrari et al. (2021). However, as stressed by the authors of the latter, the `ocUpdate` algorithm, due to its compactness, does not have the robustness and stability of the MMA algorithm. Therefore, it is recommended here to use the globally convergent MMA algorithm (Svanberg, 2007) by requesting the code from the author instead of using the `ocUpdate` if possible.

4.1.15 Continuation scheme on penalization and sharpness factor

The continuation scheme is applied to penalty and sharpness factors. To do so, the incremental increase is applied to the sharpness factor (`beta`) and penalization factor for the stiffness matrix.

```
[penalKuu ,beta] = deal(cnt(penalKuu ,
    penalCnt,loop),cnt(beta,betaCnt,loop));
```

However, to keep the relations between penalization factors as they have been mentioned in previously published codes (Homayouni-Amlashi et al., 2021) , the ratios between penalization factors (`penalratio_up` and `penalratio_pp`) which were defined before are utilized.

```
penalKup=penalKuu*penalratio_up; penalKpp=
    penalKuu*penalratio_pp;
```

4.1.16 Presentation of results

The optimization result for each iteration are presented numerically and displayed graphically. For the graphical representation, the `Display` add-on function is used. As can be seen in Fig. 7, the results of the optimization are presented in a 3D-rendered version and the 2D layout version. In 3D-rendered result, the obtained topology rendered by the iso-surfaces and 3D polarization profile are demonstrated. The passive and active materials are shown with different colors. The 2D part demonstrates the topology and polarization profile layer by layer. In this case, one can see the result of optimization for each layer. However, based on the developed code, the 2D result will be shown only if the number of layers in the $\vec{z}$ direction is less than 6 layers. For more number of layers, the 2D representation of the result will be eliminated due to the complexity of showing the results for a high number of layers and the polarization profile will be shown using the cloud of points as can be seen in Fig. 8.

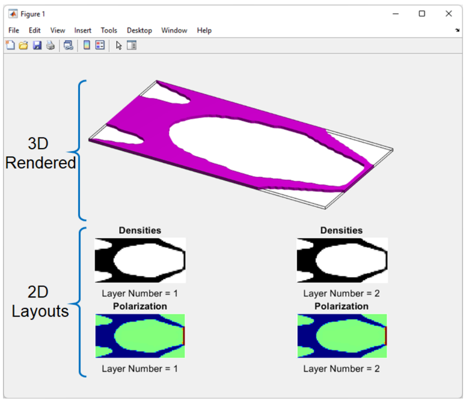

Fig. 7 Graphical representation of results by the proposed "Display" add-on function for layer based results (less than 6 layers in the thickness direction).

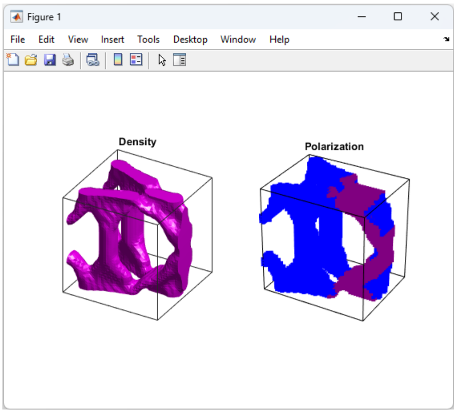

Fig. 8 Graphical representation of results by the proposed "Display" add-on function for 3D results.

4.1.17 Deformation plot

This part of the codes is provided to show the deformed topology under the application of external voltage or external force by calling the `Deformation` add-on function. In the actuation code, the response of the system under the application of voltage is recalculated by eliminating the attached spring. To do so, the global stiffness matrix is recreated by removing the stiffness of the spring. Then, the developed `Deformation` code as an add-on function is used to show the deformation of the structure.

```
Kuu = sparse(Iar( :, 1 ),Iar( :, 2 ),sK(:)); % Global stifness matrix
Kuu = Kuu+Kuu'-diag(diag(Kuu));
Ktot = [Kuu(freedofs,freedofs),Kup(freedofs,fn);Kup(freedofs,fn)',-gamma*Kpp(fn,fn)];
U =  Ktot \ [-Kup(freedofs,en)*Up(en,:);gamma*Kpp(fn,en)*Up(en,:)];  % Mechanical displacement
Uu(freedofs)=U(1:length(freedofs));
Deformation(Uu,xPhys,nelz,nelx,nely,edofMat,ElNum)
```

4.2 Energy Harvesting

The default energy harvesting code in appendix is written for the case study of Fig. 15 which contains active and passive material and is excited at the base. The general structure of the energy harvesting code is similar to actuation. As such, here the non-similar parts will be explained.

In the part `GENERAL DEFINITIONS`, the particular variables are `wj` which is the weighting function in the objective function (47),`omega` which is the excitation frequency and `Mass` which is the overall mass of attachment. In the part of `PREPARE FINITE ELEMENT ANALYSIS`, the mass matrices (`m_p,m_s`) for active and passive materials are exported from the (`FEM`) add-on function to perform a dynamic analysis.

4.2.1 Force definition

In the energy harvesting code, the procedure to define the externally applied force is generally similar to defining the desired displacement in the actuation code. The difference is that the energy harvesting code can consider several load cases by default. In this case, `nf` defines the total number of load cases. `F` is the mechanical force vector which has the equal number of columns as the number of load cases (`nf`). For each load case, the sequence of defining the number of elements, nodes and mechanical DOFs will be followed. This possibility of having one or more load cases will be discussed in the numerical examples. To build the global FEM equation (29), the total force vector (`F`) contains the externally induced charge which will be considered as zeros that is augmented to the mechanical force vector.

4.2.2 Definition of attachment mass

In the energy harvesting code, the possibility of having an attachment mass is considered by default in the code. The chosen strategy to model the attached mass is to consider some elements in the design domain heavier than the rest of the elements. This strategy can be seen in Fig. 9. With this strategy, the attached mass will be considered as a lumped mass and the geometry of the attachment will not be considered.

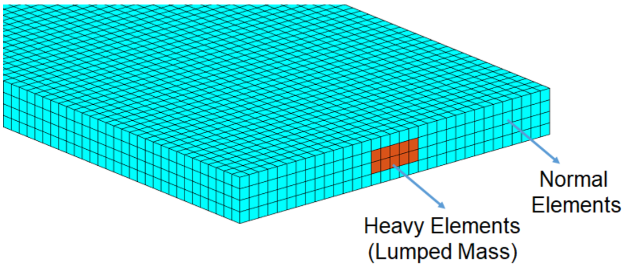

Fig. 9 Definition of attachment mass by considering heavy elements

The attachment mass is then augmented to the global mass matrix. To do so, with the same method of building the global mass matrix, we build a matrix with equal size with nonzero elements in place of the heavy elements.

```
%% DEFINITION OF ATTACHMENT MASS
sMass=zeros(nele,1);
sMass(ElNum([2:nelz-1],nelx,ceil(0.4*nely):ceil(0.6*nely)))=1; % Distribution of mass
le = Lp/nelx; we = Wp/nely; he = Ha; % Dimension of each element
ro_M = Mass*1e-3/(le*we*he)/length(find(sMass)); % Density of heavy elements
sMMass = (ro_M/ro_p)*mp_LT(:).*sMass';
sMMass = reshape( sMMass, length( mp_LT(:) ) * nele, 1 );
M_Att = sparse(Iar( :, 1 ),Iar( :, 2 ),sMMass(:)); % Mass matrix containing only the attachement mass
```

The matrix `sMass` is having rows equal to the number of elements. Among them, we chose the heavy elements that we want to consider as the attachment mass. we want to distribute this mass equally to all the selected elements. The whole procedure will be done by `sMass(ElNum([2:nelz-1],nelx,ceil(0.4*nely):ceil(0.6*nely)))=1`. Then, the global mass matrix (`M_Att`) containing only heavy elements will be built by using the indexing vectors (`Iar`).

In the optimization part of energy harvesting code, the mass matrix is built as well. The creation of the mass matrix is similar to the stiffness matrix following the interpolation function in equation (38).

```
sM = ones(length(mp_LT( : )),1).*xPhys(:)';
sM(:,Active_el)= mp_LT( : ).* sM(:,Active_el);
sM(:,Passive_el)= ms_LT( : ).* sM(:,Passive_el);
```

```
sM = reshape(  sM, length( mp_LT(:) ) * nele, 1 );

M= sparse(Iar( : , 1 ),Iar( : , 2 ),sM); % Global Mass matrix
```

It should be noted that the attachment mass is augmented to the global mass matrix after multiplication by the excitation frequency and together they form the dynamic stiffness matrix,

```
Kuu = Kuu-(M+M_Att)*omega;Kuu = Kuu+Kuu'-diag(diag(Kuu)); % Global dynamic stiffness matrix
```

The response of the system is derived using the equilibrium equation (35) and the application of equipotential condition in equation (33),

```
Ktot = [Kuu(freedofs,freedofs),Kup(freedofs,fn),Kup(freedofs,en)*B(en,:);
Kup(freedofs,fn)',-gamma*Kpp(fn,fn),-gamma*Kpp(fn,en)*B(en,:);
B(en,:)'*Kup(freedofs,en)',-gamma*B(en,:)'*Kpp(fn,en)',-gamma*B(en,:)'*Kpp(en,en)*B(en,:)];
U =  ( Ktot \ Ftot); % Response of the system
Up(fn,:) = U(lf+1:lf+length(fn),:); Up(en,:) = B(en,:)*U(lf+length(fn)+1:end,:);
```

4.2.3 Objective function and sensitivity analysis

The sensitivity analysis for energy harvesting starts by calculating the adjoint vectors using the equation (51).

```
lambda1 = zeros (ndof,nf);lambda2 = zeros (ndof,nf);mu1 = zeros (ndofPZT,nf);mu2 = zeros (ndofPZT,nf);
ADJ1 =  Ktot\[-Kuu(freedofs,freedofs)*U(1:lf,:);zeros(length(fn),nf);zeros(Nelec,nf)]; % First adjoint vector
lambda1(freedofs,:) = ADJ1(1:lf,:); mu1(fn,:) = ADJ1(lf+1:lf+length(fn),:);mu1(en,:) = B(en,:)*ADJ1(lf+length(fn)+1:end,:);
ADJ2 =  Ktot\[zeros(lf,nf);-gamma*Kpp(fn,fn)*Up(fn,:)-gamma*Kpp(fn,en)*Up(en,:);-gamma*B(en,:)'*Kpp(en,en)*Up(en,:)-gamma*B(en,:)'*Kpp(en,fn)*Up(fn,:)]; % Second adjoint vector
lambda2(freedofs,:) = ADJ2(1:lf,:); mu2(fn,:) = ADJ2(lf+1:lf+length(fn),:);mu2(en,:) = B(en,:)*ADJ2(lf+length(fn)+1:end,:);
```

The response to each adjoint equation is divided into two vectors (`lambda1, lambda2, mu1, mu2`) to separate the mechanical (λ) and the electrical part (μ) from each other. To calculate the sensitivities, the energy harvesting code considers several load cases (force) by default. Therefore, a numerator `i` is considered for each force that varies from 1 to the total number of forces (`nf`).

The objective function for the energy harvesting case includes mechanical and electrical energies which are calculated in the code and saved in the `Wm` and `We` variables. The procedure to calculate the elemental sensitivities in the energy harvesting code follows the equations (50, 60). In the written codes, the sensitivities for active and passive elements are calculated accordingly.

4.2.4 Plot deformation

In the section of `Plot deformation`, there is no modeled spring and therefore it is not needed to build the stifness matrix. Howevere, for the particular case study of the provided code, since the excitation is at base, the mechanical deformation of every nodes in the design domain should be calculated based on the mechanical displacement of the base. This is done by following line of code

```
Deformation(Uu,xPhys,nelz,nelx,nely,edofMat,ElNum) % Plot the deformation
```

In other case studies where there is no base excitation, the aforementioned line of code can be eliminated.

After explaining different parts of the energy harvesting code which are different from the actuation code, the add-on functions which will be used by both codes will be explained in the next section.

4.3 Add-on functions

There are five add-on functions that are used by both codes. These five add-on functions can be added to the end of each code to make it solely independent from calling external functions to compile. The ingredients of each add-on function will be explained in the upcoming sections.

4.3.1 Deformation

This function produces the elemental discretization of the optimized topology with and without deformation based on the vector of displacement. Since the structural deformation is small according to the dimension of the structure, the displacement vector is normalized by factor `AMP` which amplifies the deformation. This amplification factor can be modified for a better demonstration of the result.

4.3.2 Matrix Rotation

This function produces the new tensor matrices based on the polarization direction and rotation of the local coordinates. The implementation code is based on equations (3)-(6). Three polarization directions are considered by default in the directions of the coordinates system. It is possible to consider other directions as well by re-calculating the Euler angles.

4.3.3 Display

The display function plots the results in 2D and 3D formats. In the 3D representation of the result, two different colors are considered for active and passive material. In addition, a transparent box with highlighted edges shows the borders of the design domain. If the number of elements in the $\vec{y}$ direction is less than 6, then there will be a 2D representation of the result as well. In the 2D part, the topology layout and polarization profile of each layer are illustrated.

4.3.4 Finite element matrices (FEM)

The FEM add-on function builds the finite element matrices. This function gets the number of elements in each direction, the geometrical dimension of the design domain and the tensor matrices. With the geometrical dimension of the design domain and the resolution of the mesh, the geometrical dimensions of the element can be obtained (`le,we,he`). After defining the dimensions of the elements, the function is separated to two parts based on the elements type (`EL_T`).

In the trilinear element section, the two point Gauss quadrature method is used to solve the integration in equation (20), the Gauss points are stored in (`GP`). By using the 8-node rectangular element which is a particular form of the hexahedron element, there will be a simplification in the Jacobean matrix (`J`) and its determinant (`detJ`). The natural coordinates ξ, η and μ are defined in the code as `s`, `t` and `u` respectively. The matrix `DN` is the derivation of the interpolation functions with respect to the natural coordinates. Based on equations (17) and (18), we build the matrix B_ϕ and then the matrix B_u can be obtained from the elements of matrix B_ϕ. Finally, all the elemental FEM matrices (`kuu, kup, kpp, m`) are built by following lines of code

```
kuu = kuu + transpose(Bu)*C*Bu*detJ; % Stiffness matrix
kup = kup + Bu'*e'*Bphi*detJ;  % Piezoelectric coupling matrix
kpp = kpp + Bphi'*Ep*Bphi*detJ; % Dielectric stiffness matrix
m = m+detJ*ro*(N'*N); % mass matrix
```

To build the quadratic element matrices, the general procedure is similar to the trilinear elements. However, the three point Gauss quadrature method is used to solve the integration in equation (20) and the interpolation functions are derived using the equations (22)-(25).

Based on the chosen type of element, the FEM function produces outputs containing the number of nodes in each element `EL_NN`, total number of nodes in the design domain which is equivalent to the total number of electrical DOFs `ndofPZT` and finally internal node numbers of 6 faces of element `TOPNODS,BOTNODS,FRNODS,BAKNODS,LEFNODS,RTNODS`.

4.3.5 OC update algorithm

The `ocUpdate` algorithm is a routine to update the optimization variables written by Ferrari et al. (2021). The same lines of codes are used here and for the sake of brevity, readers are referred to aforementioned paper to understand the developed code.

In this section, different parts of the code are explained in detail. In the upcoming section, we will explain how the codes can deal with different case studies similar to ones that exist in the literature.

5 Numerical Examples

In this section, the goal is to investigate the performances of the developed codes in different case studies of actuation and energy harvesting. Each case study can be implemented by few modifications to the original codes provided in the appendix. For the sake of simplicity, we provided separate codes for each case study as supplementary material to this paper.

5.1 Actuation

5.1.1 Bi-morph actuator

In this part, we start with the bi-morph piezoelectric actuator which is shown in Fig. 10-(a). The general goal of the bi-morph actuator is to push a working object in a specified direction. In the first case study, the idea is to have a bi-morph piezoelectric bending actuator with the desired direction of displacement as can be seen in Fig. 10-(a). As it was mentioned before, a mechanical spring with stiffness `Ks` is modeled to simulate the stiffness of the working object. The Matlab code for actuation which is provided in the appendix is written for this example. In this code, quadratic element is chosen by default (i.e. `EL_T=2`).

The result of optimization can be seen in panel (b) of Fig. 10 in 3D-rendered format. The deformation of the structure due to the application of the voltage is demonstrated in panel (c). Panel (e), demonstrates the density and polarization profiles for each layer. The value of the objective function in each iteration is shown in panel (d). We stopped the iterations after 200 iterations since there was no change in the obtained layout. It should be noted that the user can change the stiffness of the modeled spring to obtain other optimized topologies. The optimized polarization profile is uniform with the same sign of polarity. However, this similarity of polarization sign is not equivalent to similar contraction or expansion of both layers. Since the middle electrode is grounded, the same polarity sign means one layer will expand and the other layer will contract which results in bending out-of-plane (transverse direction).

In appendix, it is demonstrated that trilinear elements cannot be used for low thickness to length ratio. We investigated the result produced by the two types of elements for the first case study of the piezo bending actuator. The result can bee seen in Fig. 11 for different thickness. For linear element, we also considered four elements in direction of thickness (2 element per layer). As can bee seen in Fig. 11, just for very low thickness of the piezoelectric plate the results from linear element will be different to quadratic element. This is in correspondence to conclusion made in the validation of the FEM in appendix where for very low thickness layer, the error of linear elements are significant due to "shear lock" phenomenon Cook et al. (2007).

The next case study, is a bi-morph piezoelectric gripper as illustrated in Fig. 12. The MATLAB code CASE2.m is provided as supplementary material to this paper for the implementation of this case study. This case study, has been investigated in the authors previous published codes as well (Homayouni-Amlashi et al., 2021). In this case study, the goal is to grip an object in the desired space. This case study is defined by considering half of the design domain using the symmetry and defining the roller boundary condition in the symmetry line as illustrated in the mechanical interpretation of the problem in Fig. 12. The optimized result in rendered form and the produced deformation is illustrated in panels (b) and (c) of the figure. Panels (e-h) show the 2D layouts and the optimized polarity. The polarization profile demonstrates how the algorithm uses the combination of expansion and contraction to produce the gripping task.

5.1.2 3D Piezoelectric actuator (combination of active and passive material)

In case studies 3 and 4, the goal is to consider 3D piezoelectric actuators that consist of active and passive domains. To start with, we can consider the geometrical sketch of the desired 3D actuator as in Fig. 13-(a). In this figure, a rectangular domain of passive material is sandwiched between two piezoelectric plates. The desired direction of the displacement demonstrates that the target is designing a pusher. The implementation MATLAB codes (CASE3.m and CASE4.m) are provided as supplementary materials.

The results of the optimization and the deformation of the structure due to voltage application are illustrated in Fig. 13-(b) and (c) respectively. As it is obvious from these figures, the bending deformation of the top and bottom piezoelectric plates will result in the deformation of the desired point in the $\vec{x}$ direction.

The optimization problem in panel (d) is very similar to the previous case study but with four patches of piezoelectric plates mounted by pairs at the top and bottom of a cuboid passive material.

The case study number (5) as illustrated in Fig. 14 is similar to case studies (3) and (4). The difference is that the design domain consists off the active material. The optimization method is applied on a 3D piezoelectric block which is sandwiched between two electrodes on top and bottom.

The rendered optimization result, polarization profile and produced deformation are illustrated in panels (b-c) of Fig. 14. By inspecting the polarization profile, we can observe that a combination of expansion and contraction produces the desired deformation.

5.2 Energy harvesting

5.2.1 Multi-morph energy harvester with tip attachment

The next case study is the energy harvesting with multilayer piezoelectric cantilever beam with tip attachment which will be excited at the base. This case study is the most widespread one in the literature for piezoelectric energy harvesting. The base code for energy harvesting in the appendix is written for this case study and its geometrical sketch can be seen in Fig. 15-(a). Based on the code, it is obvious that the total number of layers in the $\vec{z}$ direction is considered to be four in which the top and bottom layers are considered to be piezoelectric materials and the two middle layers are considered to be passive material. Aluminum is chosen as the passive material and PZT-4 as the piezoelectric material. In the

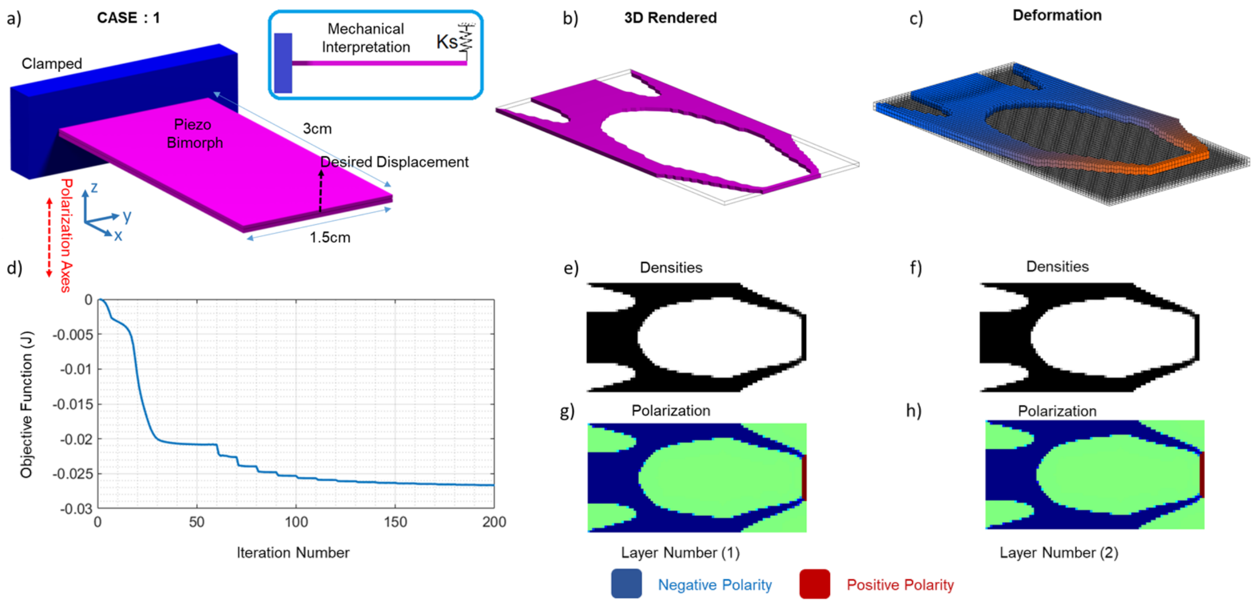

Fig. 10 Topology optimization of a piezoelectric bending actuator. Case 1: a) Geometrical sketch with mechanical interpretation. b) 3D-rendered optimized topology. c) Deformation plot using the discretization elements, d) Objective function, e) 2D layouts and polarization profiles

optimization, both the passive and active material will be optimized. The rendered result and the respective deformation are plotted in panels (b) and (c). In this case study, we also consider other piezoelectric materials to inspect the changes in the result. To do so, three other piezoelectric materials including Aluminum Nitride (ALN), Lithium Niobate (L-NIO) and PVDF are considered and the results of optimization have been shown in panels (d-f). The numerical results regarding mechanical and electrical energies are also reported in panels (g-h).

As illustrated in Fig. 15, It is obvious that by changing the piezoelectric material the optimized topology will be different. This difference can mostly be seen in the optimized passive material. Hence, for each result, the optimized passive material is shown separately as well. Moreover, from the plots of Fig. 15-(g-h), it seems that PZT-4 has a high electrical to mechanical energy ratio. This is expected since the electromechanical coupling coefficient for PZT are higher than others.

In the case study (7) as illustrated in Fig. 16, we consider a case in which a piezoelectric block is sandwiched between two layers of passive material and the polarization axes is in the $\vec{x}$ direction. Indeed, in this case, the piezoelectric energy harvester will work in shear-mode benefiting the high e15 coupling coefficient. This is the same scenario which is studied in the paper of Malakooti and Sodano (2015). The e15 coupling coefficient is higher than the other coupling coefficients in most of the piezoelectric materials of the 4mm crystal class. The illustrated result in Fig. 16 -(b-c) is for one `wj`. Obtaining more efficient result by changing the optimization parameters is up to the readers.

5.2.2 3D energy harvester

In the next case study (8), we consider a 3D design domain for the energy harvester under the application of two harmonic load cases as has been shown in Fig. 17-(a). Indeed, this design domain is inspired by the proposed 3D structure in (Fattahi and Mirdamadi, 2020, 2019).

In Fig. 17, four different topologies of the 3D energy harvester are shown for 4 different values of weighting factor `wj`. As can be seen in this figure, by reducing the weighting factor, the algorithm tries to reduce the stiffness of the passive material to increase the electrical output energy. However, too much reduction of the `wj` will result in divergence of the results where the mechanical and electrical energies will start to increase as can be seen in plots in panels (h and i) of the figure. This is due to no weight on the mechanical performance of the structure. This case study is provided to demonstrate the ability of the provided code to deal with 3D structural domains as well as the effects of the weighting factor `wj`.

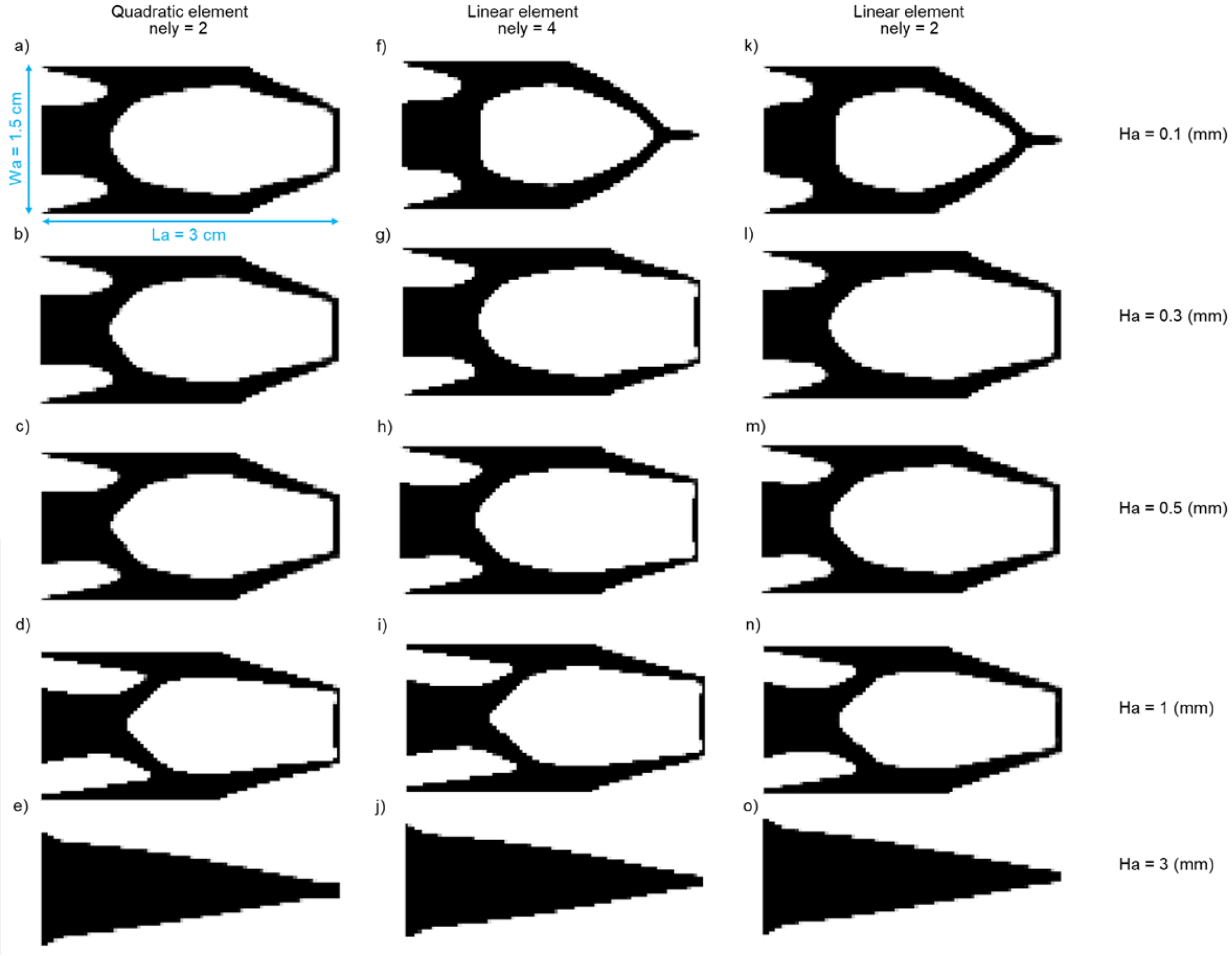

Fig. 11 Topology optimization results of piezoelectric bending actuator for different thickness of the piezoelectric layer and different element types.

6 Discussion

The case studies provided in the numerical sections demonstrate the capabilities, flexibility and efficiency of the provided codes to deal with different scenarios. It is worth mentioning here that the goal in the numerical section was not to provide the best result. It is left to the readers to find the best parameters of optimization.

The provided codes in the appendix are geometry based. In fact, this contrasts to most of the published codes in the literature, which are normalized based on the geometry and physical properties. In this paper, users should carefully choose the geometry of the design domain and resolution of mesh independently. Inappropriate choice of geometry of active and passive material and mesh resolution may lead to meaningless results.

Fabrication and realization of piezoelectric materials are very challenging which limits the production of complicated geometries of these materials. This limitation is not considered in this paper and augmentation of manufacturing constraint can be considered as a future target.

7 Conclusion

Two MATLAB implementation codes are proposed for the application of SIMP topology optimization to the piezoelectric actuators and energy harvesters using the 3D finite elements. A detailed finite element modeling with providing two types of linear and quadratic elements has been presented for the design domain which contains a combination of passive and active materials. The SIMP topology optimization is extended by using the PEMAP-P strategy to optimize the topology and polarization profile at the same time. The implementation codes are written in the most flexible and user-friendly format. A new indexing method is proposed for numbering the elements, nodes and mechan-

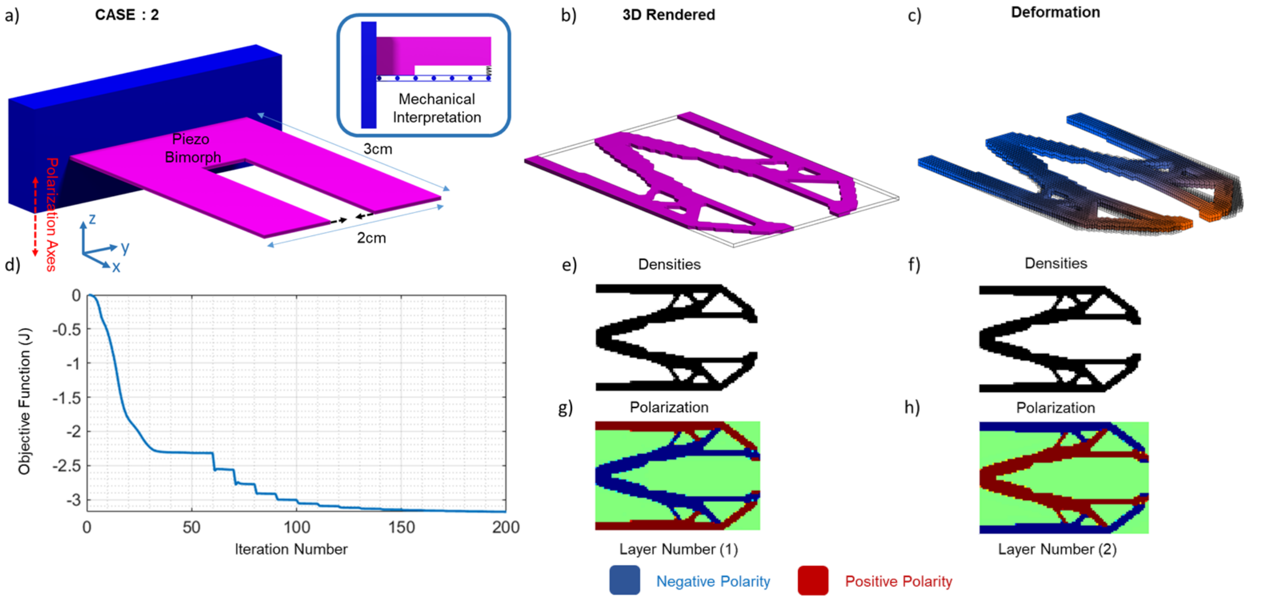

Fig. 12 Topology optimization of a piezoelectric gripper. Case 2: a) Geometrical sketch with mechanical interpretation. b) 3D-rendered optimized topology. c) Deformation plot using the discretization elements, d) Objective function, e) 2D layouts and polarization profiles

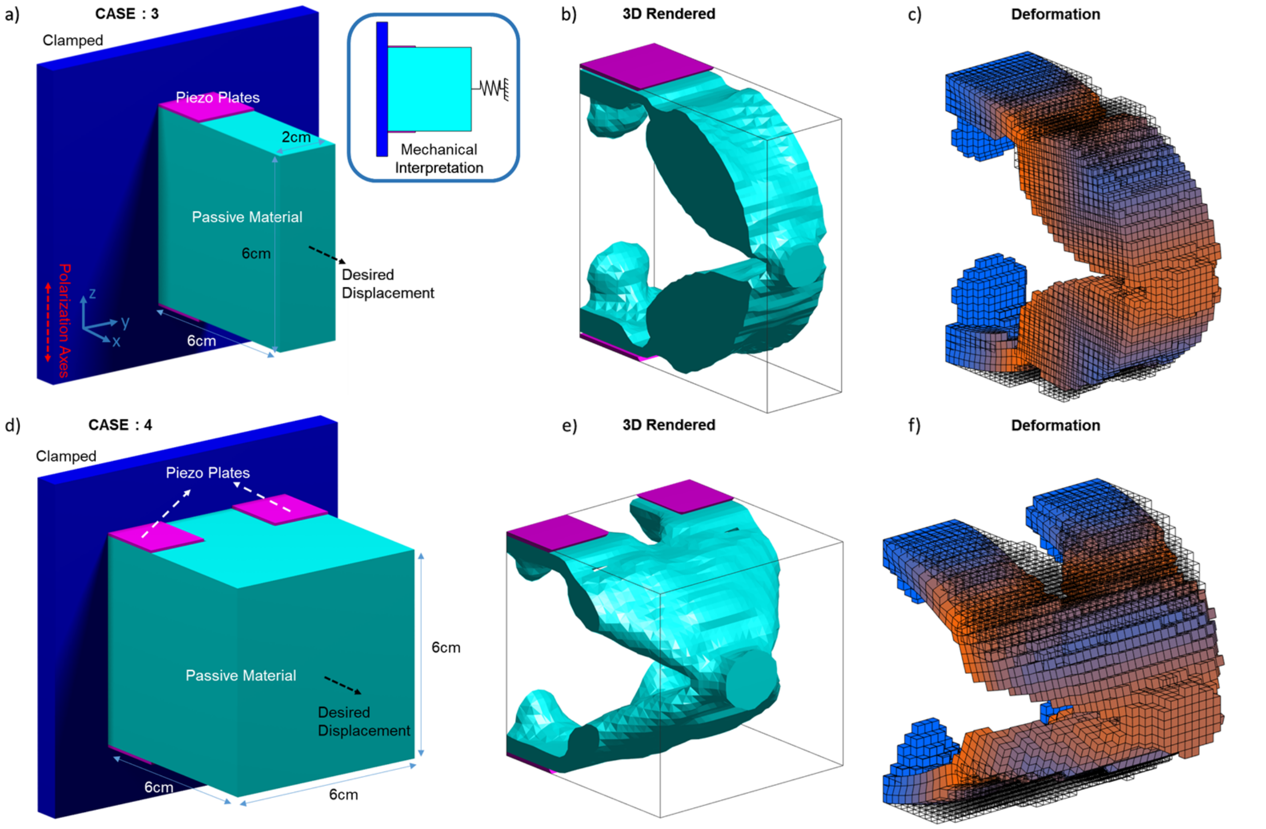

Fig. 13 Topology optimization of 3D Piezoelectric actuators. Case 4: a) Geometrical sketch with mechanical interpretation. b) 3D-rendered optimized topology. c) Deformation plot using the discretization elements, Case 5: d) Geometrical sketch. e) 3D-rendered optimized topology. f) Deformation plot using the

ical DOFs which facilitates the definition of boundary conditions, electrodes, etc. Different parts of the imple-

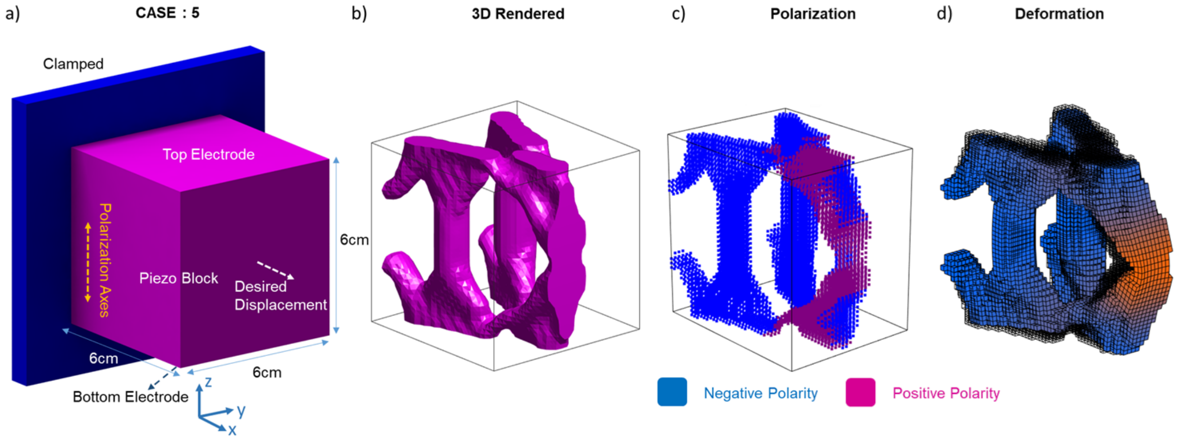

Fig. 14 Topology optimization of 3D piezoelectric pusher. Case 5: a) Geometrical sketch. b) 3D-rendered optimized topology. c) Polarization profile, d) Deformation plot using the discretization elements

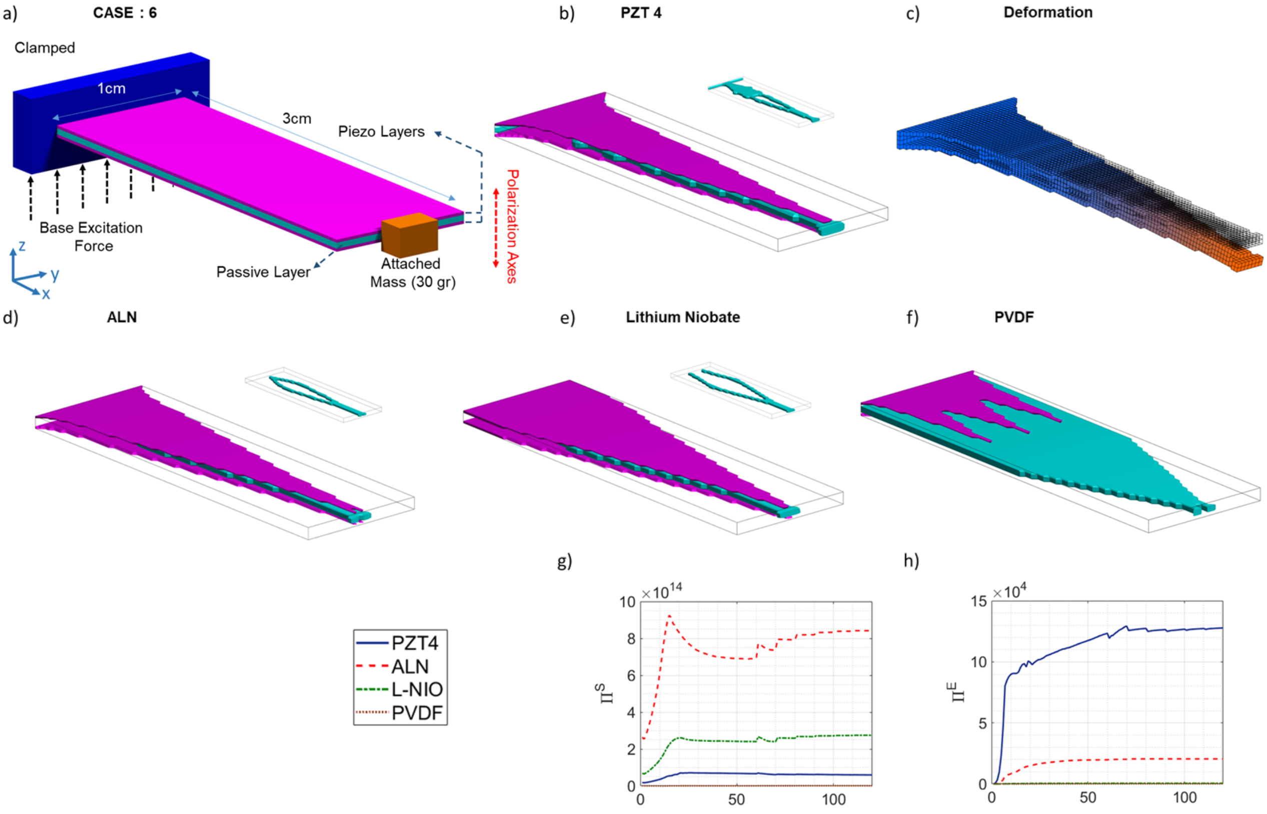

Fig. 15 Topology optimization of a multi-layer piezoelectric energy harvester with tip attachment for different piezoelectric materials. Case 6: a) Geometrical sketch with mechanical interpretation. b) 3D-rendered optimized topology (PZT-4). c) Deformation plot using the discretization elements, Case 5: d),e),f) Rendered results for different piezoelectric materials. g) Mechanical energy plots, h) Electrical energy plots.

mentation codes are explained in detail to be comprehensive for newcomers in the field. Several case studies are provided to demonstrate the efficiency of the codes in solving practical optimization problems in actuation and energy harvesting modes. Future work would consider the fully multi-material optimization considering random distribution of piezo and non-piezo materials as well as consideration of manufacturing constraints.

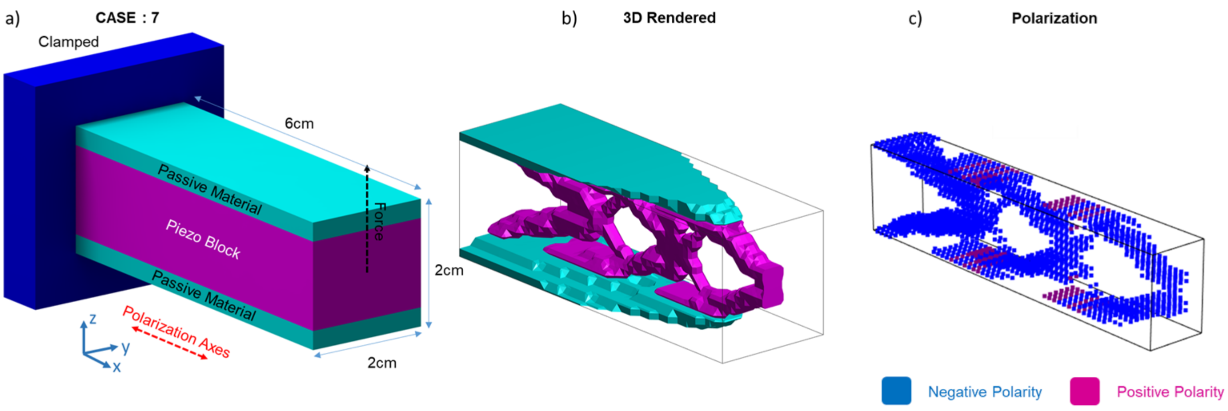

Fig. 16 Topology optimization of a multi material piezoelectric energy harvester working in shear-mode under application of a harmonic load. Case 7: a) Geometrical sketch, b) 3D-rendered optimized topology, c) Polarization profile

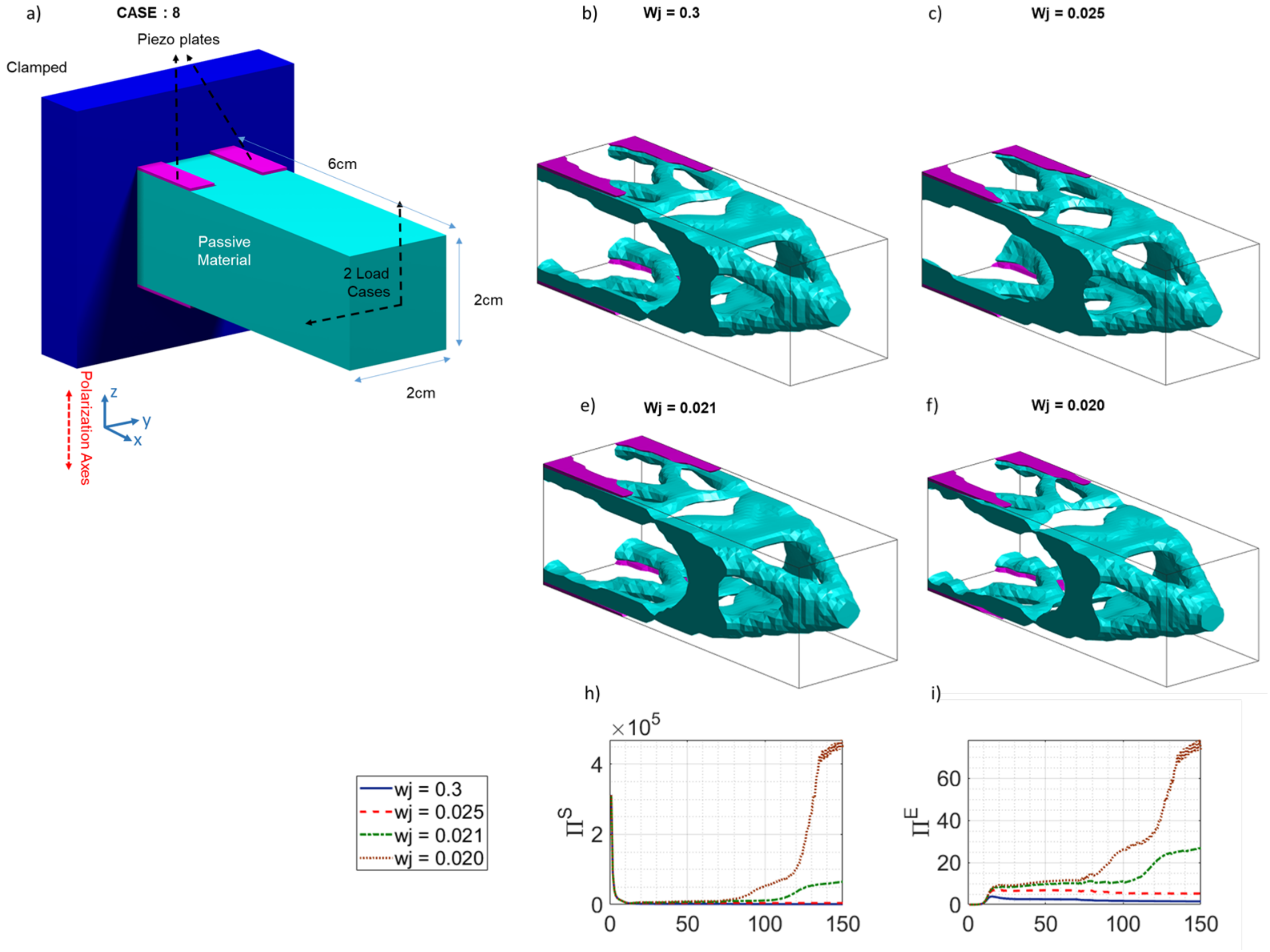

Fig. 17 Topology optimization of a 3D piezoelectric energy harvester under application of two load cases. Case 7: a) Geometrical sketch, b),c),e),f) 3D-rendered optimized topology for different weighting factors, h) Mechanical energy plots, i) Electrical energy plots

Declarations

Acknowledgement of fundings: This work has been supported by 1- The national CODE-TRACK project (ANR-17-CE05-0014-01, Control theory tools for optimal design of piezoelectric energy harvesters devoted to birds tracking devices), 2- OptoBots project (ANR-21-CE33-0003), 3- EIPHI Graduate School (contract ANR-17-EURE-0002) 4- Villum Investigator project AMSTRAD (VIL54487) from VILLUM FONDEN

Conflict of interest: The authors declare that they have no conflict of interest

Replication of results: All the results presented in this work can be reproduced with the MATLAB codes available within the paper.

8 Appendix

8.1 Validation of the finite element method with COMSOL Multiphysics

In this section, we try to investigate the accuracy of the FEM part of the paper in modeling a piezoelectric plate actuator by comparing the results with with COMSOL Multiphysics software. In the developed FEM of the paper, we considered two types of elements: trilinear elements and quadratic elements.

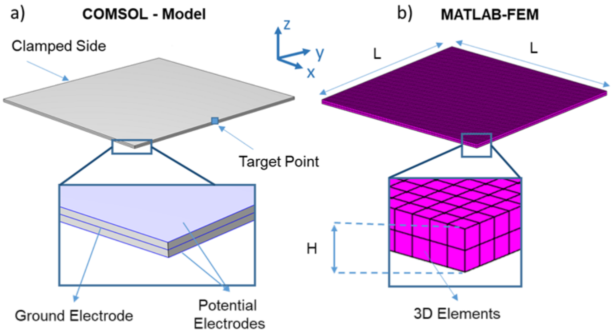

Fig. 18 Modeling and square two layer piezoelectric plate in a) COMSOL and b) MATLAB

It is already known in the literature that trilinear elements have a deficiency called ”shear locking”. This term corresponds to a parasitic shear strain which is explained by Cook et al. (2007). Due to this deficiency, trilinear elements cannot exhibit bending. To solve the problem of shear locking we have two solutions: 1- We can use the B-Bar method suggested by Bower (2009) which modifies the strain displacement matrix, 2-We can use the quadratic elements. In this paper, we chose the latter for their generally higher precision. Moreover, it will be interesting to investigate the accuracy of the developed FEM with trilinear and quadratic elements and to see when we are obliged to use quadratic elements and when it is possible to use the trilinear element and save the computation time. In this regard, we considered a square two-layer piezoelectric plate as shown in Fig. 18, which will be actuated by applying potentials on the electrodes to produce a bending deformation in the $\vec{z}$ direction. For this configuration, the FEM responses from developed MATLAB codes of linear and quadratic elements are compared with the result of COMSOL. To do so, the displacement of a target point at the tip of the piezoelectric plate as shown in Fig. 18 is calculated and compared between different platforms.

In COMSOL, the two-layer piezoelectric plate with the same geometrical dimension is modeled. To discretize the design domain the built-in tetrahedral elements are employed. To define the resolution of mesh, the “fine” option has been chosen where COMSOL automatically choose the proper dimension of the element. We didn’t use the fixed dimensions for different case studies since by changing the thickness of the piezoelectric plate, small size of elements makes the computation time huge. Moreover, by few inspections with considering finer mesh, the change in the result was negligible.

The numerical investigation of the displacement of the target point is provided in Fig. 19. In panel (a), it has been shown that the thickness of the piezoelectric plate is changed while the resolution of the mesh remains constant. One element per each piezoelectric layer is considered and the thickness is changed to see the accuracy of FEM model using different elements. As can be seen in panel (b), the response of the MATLAB FEM with quadratic elements are in fine agreement with the results of COMSOL. However, the response from MATLAB FEM with trilinear elements lacks the accuracy specially in low thickness piezoelectric layers. Independent from the thickness of the piezoelectric layers, the error of the quadratic elements are below one percent as illustrated in panel (c).

The error of the trilinear element start from enormous value of 40 percent for 0.2mm thickness and then reach to 6 percent for 2 mm of thickness which is equivalent to 0.033 thickness to length ratio for each piezoelectric layer. Since the results of quadratic elements are quite satisfactory, in the next panels of Figure (3), we just investigate the conditions to improve the results for trilinear elements. In panel (d), for the 0.2 mm thickness, we increased number of elements in the direction of thickness (i.e. nelz). It can be seen in panels (e) and (f) that increasing nelz can not increase the accuracy significantly. In panel (g), the same effect of increasing nelz is investigated with H=6 mm. As can be seen the error is much less than H=0.2, However, still the effect of increasing nelz is not significant. Just increasing the nelz from 2 to 4, decrease the error from 6 to 2 percent. In panel (j-o), we investigated the increase of number of elements in the $\vec{x}$ and $\vec{y}$ direction (i.e. nelx and nely). As it can be seen in panels (k) and (l), increasing the resolution of the mesh in the $\vec{x}$ and $\vec{y}$ direction can significantly increase the accuracy. It can be concluded that for low thickness layers, high resolution of mesh

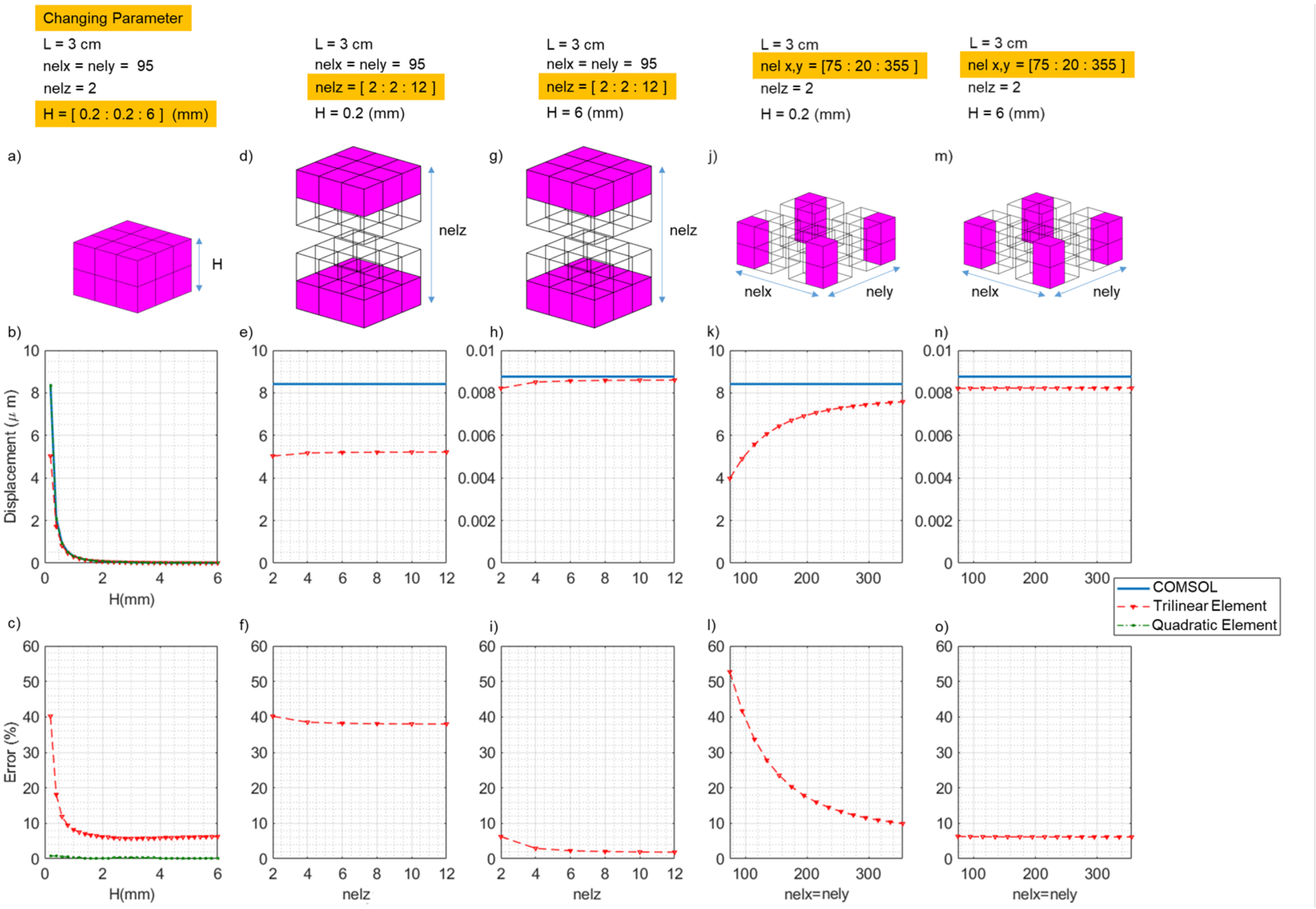

Fig. 19 Validation of FEM modeling with COMSOL multiphysics software

is necessary. However, the error percentage will not be less than 10 percent, which is still enormous. For thick piezoelectric plate in panel (n) and (o) it is obvious that increasing the resolution of mesh in the $\vec{x}$ and $\vec{y}$ direction will not improve the accuracy.

The overall conclusion of the aforementioned investigation is that, quadratic elements are quite accurate even in coarse discretization mesh. On the other hand, trilinear elements are very weak in terms of capturing the bending of thin piezoelectric plate i.e. thickness to length ratio less than $1/30 = 0.033$. However, for larger values of thickness to length ratio, a reasonable definition of mesh resolution and definition of two elements per each layer can bring the error in the scale of two percent.

To check the generality of FEM accuracy, we change the geometry of the plate by decreasing the length of the piezoelectric plate (L) to 1cm. Then, we performed the same analysis of figure (3) but with different values of thickness. By inspecting the plats in figure (4), it can be seen that the same behavior for the thickness to length ratio of the plate can be seen in the accuracy of different elements. In panel (b) and (c), it can bee seen that by increasing the thickness of the of the piezoelectric layers the accuracy of the linear elements, increases. For 0.5 (mm) of thickness which is 0.05 thickness to length ratio the error converges to 6.2 percent. Then, by increasing the number of nelz to two for each piezoelectric layer, the error decreases to 2 percent. This is the similar result for thickness to length ratio that we had in figure (3). In fact, the plots in figure (4) are almost similar to plots in figure (3) and same analysis can be provided by both. It should be noted that the accuracy of quadratic elements are quite satisfactory again for all thickness to length ratios.

As a conclusion, when the total thickness to length of the beam structure is less than 0.05, the quadratic elements must be used to discretize the design domain. For higher thickness to length ratio, trilinear elements can be used by considering 2 elements in the direction of thickness.

8.2 Piezoelectric Material Properties

Material properties for the piezoelectric materials which are used in this paper are mentioned in Table 1. All

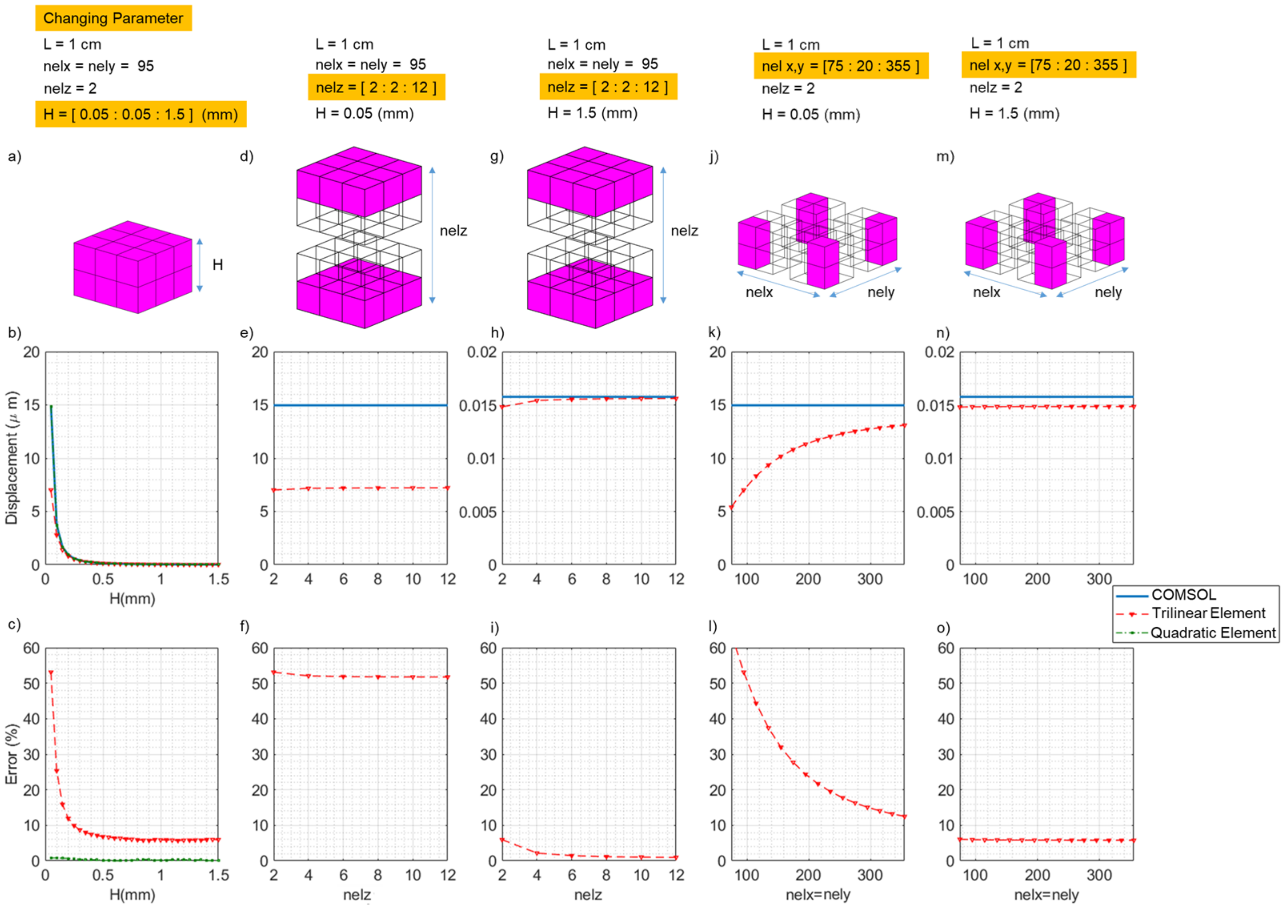

Fig. 20 Validation of FEM modeling with COMSOL multiphysics software

the material properties in this paper are extracted from COMSOL multiphysics software version 6.1. The reported piezoelectric coefficients are in stress-charge form. However, sometimes in data sheets of piezoelectric materials, the coefficients are reported in the strain charge form. In this regards, the calculation of the stress-charge coefficients from the strain-charge coefficients are as follows

$$c^E = (S^E)^{-1}, \quad e = d(S^E)^{-1}, \quad \varepsilon^S = \varepsilon^{\overline{T}} - d(S^E)^{-1}d^T \tag{60}$$

where S^E is the strain tensor in constant electrical field, d is the coupling matrix in strain-charge format and $\varepsilon^{\overline{T}}$ is the permittivity in constant stress.

Table 1 Piezoelectric material properties

	Coefficicents of Elasticity Matrix ($\times 10^{+11}$)										
	c_{11}^E	c_{12}^E	c_{13}^E	c_{14}^E	c_{22}^E	c_{33}^E	c_{24}^E	c_{44}^E	c_{55}^E	c_{66}^E	Density (kg/m^3)
ALN	4.1	1.49	0.99	0	4.1	3.89	0	1.25	1.25	1.305	3300
LiNbO3	2.02	5.29	7.49	8.99	2.02	2.43	-8.99	5.99	5.99	7.48	4700
PVDF	0.038	0.019	0.09	0	0.038	0.012	0	0.009	0.009	0.007	1780
PZT - 2	1.34	0.678	0.680	0	1.34	1.13	0	0.222	0.222	0.334	7600
PZT - 4	1.39	0.778	0.742	0	1.39	1.15	0	0.25	0.25	0.30	7500
PZT - 5H	1.27	0.802	0.846	0	1.27	1.17	0	0.229	0.229	0.234	7500
PZT - 7A	1.577	0.876	0.812	0	1.577	1.256	0	0.294	0.294	0.349	7700
PZT - 8	1.468	0.810	0.810	0	1.468	1.317	0	0.313	0.313	0.328	7600
ZincOxide	2.097	1.211	1.05	0	2.097	2.111	0	0.423	0.423	0.442	5680

	Coefficicents of Coupling Matrix								Relative Permittivity Matrix ($\times 8.85 \times 10^{-12}$)		
	e_{15}	e_{16}	e_{21}	e_{22}	e_{24}	e_{31}	e_{32}	e_{33}	ε_{11}^S	ε_{22}^S	ε_{33}^S
ALN	-0.48	0	0	0	-0.48	-0.58	-0.58	1.55	9	7.4	9
LiNbO3	3.69	-2.53	-2.53	2.53	3.69	0.19	0.19	1.30	43.6	43.6	29.16
PVDF	0	0	0	0	0	0.024	0.001	-0.027	7.4	9.3	7.6
PZT - 2	9.77	0	0	0	9.77	-1.81	-1.81	9.05	504.1	504.1	270
PZT - 4	12.71	0	0	0	12.71	-5.20	-5.20	15.08	762.5	762.5	663.2
PZT - 5H	17.03	0	0	0	17.03	-6.62	-6.62	23.24	1704.4	1704.4	1433.6
PZT - 7A	10.58	0	0	0	10.58	-2.29	-2.29	9.48	499.5	499.5	229.9
PZT - 8	10.34	0	0	0	10.34	-3.87	-3.87	13.91	904.4	904.4	561.6
ZincOxide	-0.48	0	0	0	-0.48	-0.56	-0.56	1.32	8.544	8.544	10.204

MATLAB 3D TOPOLOGY OPTIMIZATION CODE FOR PIEZOELECTRIC ACTUATORS

```
% 3DTOPIEZO_ACTUATION // Abbas Homayouni-Amlashi et al. 2024
clc ;clear ;close all
%% GENERAL DEFINITIONS
La = 3e-2; Wa = 1.5e-2; Ha = 0.1e-2;  % Pieozoelectric geometrical dimensions (length, width, height) (m)
Lp = 3e-2; Wp = 1.5e-2; Hp = 0.4e-3;  % Passive material geometrical dimensions (length, width, height) (m)
nelx = 80; nely = 40; nelz = 2; % Number of elements in each direction
penalKuu = 3; penalKup = 4; penalKpp = 4 ;penalPol = 1; % Penalization factors
EL_T = 2; % Element type 1: trilinear 2: quadratic
volfrac = 0.4; % Volume fraction
Max_loop = 200; % Maximum number of iterations
pol_dir = 'z'; % Piezoelectric polarization direction
rmin = 3; % Filter radius
ft = 2; % 1= Density filter, 2&3= projection with eta and beta as parameters
ftBC = 'N';
eta = 0.5; % Threshold
beta = 2; % Sharpness factor
penalCnt = {60,5,10,0.2}; % Continuation scheme on penalKuu {istart, maxPar, isteps, deltaPar}
betaCnt = {60,30,10,2}; % Continuation scheme on beta {istart, maxPar, isteps, deltaPar}
move = 0.2; % Optimization variable update move
Dir = 0; % Direction of movement x=2; y=1; z=0;
Ks = 0.1; % Spring stiffness
%% MATERIAL PROPERTIES (PZT 4)
ro_p = 7500; % Density of piezoelectric material (kg/m^3)
C_p = [1.3900    0.7784    0.7428    0.0000    0.0000    0.0000
0.7784    1.3900    0.7428    0.0000    0.0000    0.0000
0.7428    0.7428    1.1541    0.0000    0.0000    0.0000
0.0000    0.0000    0.0000    0.2564    0.0000    0.0000
0.0000    0.0000    0.0000    0.0000    0.2564    0.0000
0.0000    0.0000    0.0000    0.0000    0.0000    0.3058]*1.0e+11; % Piezoelectric stiffness tensor
e   = [0.0000    0.0000    0.0000   0.0000  12.7179  0.0000
0.0000    0.0000    0.0000  12.7179   0.0000    0.0000
-5.2028   -5.2028   15.0804   0.0000   0.0000    0.0000]; % Piezoelectric coupling matrix
Ep  = [0.6746    0.0000    0.0000
0.0000    0.6746    0.0000
0.0000    0.0000    0.5867]*1.0e-08; % Piezoelectric permittivity matrix
ro_s = 2710; % Density of Passive material (kg/m^3)
EE = 70e9; % Passive material Young modulus of elasticity
nu = 0.3 ; % Poisson's ratio
C_s = (EE/((1+nu)*(1-2*nu)))*[1-nu nu nu 0 0 0 ; nu 1-nu nu 0 0 0 ; nu nu 1- nu 0 0 0 ;
0 0 0 (1-2*nu)/2 0 0 ; 0 0 0 0 (1- 2*nu)/2 0 ; 0 0 0 0 0 (1- 2*nu)/2]; % Passive material stiffness tensor
%% PREPARE FINITE ELEMENT ANALYSIS
[C_p_1,e_1,Ep_1] = Matrix_Rotation(C_p,e,Ep,pol_dir); % New tensor matrices after Rotation of coordinate system
[kuu,kup,kpp,~,ndofPZT,EL_NN,TOPNODS,BOTNODS,FRNODS,BAKNODS,LEFNODS,RTNODS] = FEM(La,Wa,Ha,nelz,nelx,nely,C_p_1,e_1,Ep_1,0,EL_T); % Piezoelectric elemental matrices
[ks,~,~,~,~,~,~,~,~,~,~,~] = FEM(Lp,Wp,Hp,max(1,nelz-2),nelx,nely,C_s,zeros(3,6),zeros(3,3),0,EL_T); % Passive material elemental matrices
k0 = max(abs(kuu(:)));beta0 = max(kpp(:));alpha0 = max(kup(:)); % Normalization Factors
kuu = kuu/k0;ks = ks/k0;kup = kup/alpha0;kpp = kpp/beta0;gamma = (k0*beta0)/(alpha0^2); % Application of normalization
kuu_LT = kuu(tril(true(size(kuu)))); % Vector of lower triangular matrix
kpp_LT = kpp(tril(true(size(kpp)))); % Vector of lower triangular element of piezoelectric dielectric stifness matrix
ks_LT = ks(tril(true(size(ks)))); % Vector of lower triangular matrix
ndof = 3*ndofPZT; % Mechanical degrees of freedom
nele = nelx*nelz*nely; % Number of elements
ElNum = reshape(1:nele,nelz,nelx,nely); % Element indexing
% Building connectivity matrix
NNlinear=(nelz+1)*(nelx+1)*(nely+1);
edg1=reshape(1:NNlinear,nelz+1,nelx+1,nely+1);
edg2=reshape(NNlinear+1:NNlinear+(nely+1)*(nelx+1)*nelz,nelz,nelx+1,nely+1);
edg3=reshape(NNlinear+(nely+1)*(nelx+1)*nelz+1:NNlinear+(nely+1)*(nelx+1)*nelz+(nelz+1)*(nely+1)*nelx,nelz+1,nelx,nely+1);
edg4=reshape(NNlinear+(nely+1)*(nelx+1)*nelz+(nelz+1)*(nely+1)*nelx+1:NNlinear+(nely+1)*(nelx+1)*nelz+(nelz+1)*(nely+1)*nelx+nely*(nelz+1)*(nelx+1),nelz+1,nelx+1,nely);
n=0;
for i=1:nely
for j=1:nelx
for k = 1:nelz
n=n+1;
EDG1=edg1([k,k+1],[j,j+1],[i,i+1]);
EDG2=edg2(k,[j,j+1],[i,i+1]);
EDG3=edg3([k,k+1],j,[i,i+1]);
EDG4=edg4([k,k+1],[j,j+1],i);
ED(n,:)=[EDG1(:);EDG2(:);EDG3(:);EDG4(:)]';
end
end
end
EDM = ED(:,[2,4,3,1,6,8,7,5,14,10,13,9,20,19,17,18,16,12,15,11]);
edofMatPZT = EDM(:,1:EL_NN); % Electrical connectivity matrix
edofMat(:,3:3:3*EL_NN)=3*edofMatPZT;edofMat(:,2:3:3*EL_NN)=3*edofMatPZT-1;edofMat(:,1:3:3*EL_NN)=3*edofMatPZT-2; % Mechanical connectivity matrix
[ sI, sII ] = deal( [ ] );
for j = 1 : 3*EL_NN
sI = cat( 2, sI, j : 3*EL_NN );
sII = cat( 2, sII, repmat( j, 1, 3*EL_NN - j + 1 ) );
end
[ iK , jK ] = deal( edofMat( :,  sI )', edofMat( :, sII )' );
Iar = sort( [ iK( : ), jK( : ) ], 2, 'descend' ); clear iK jK % Assembly indexing (stiffness matrix)
[ sI, sII ] = deal( [ ] );
for j = 1 : EL_NN
sI = cat( 2, sI, 1 : 3*EL_NN );
sII = cat( 2, sII, repmat( j, 1, 3*EL_NN  ) );
end
[ iKup , jKup ] = deal( edofMat( :,  sI )', edofMatPZT( :, sII )' );
Iar_up = [ iKup( : ), jKup( : ) ] ; clear iKup jKup; % Assembly indexing for piezoelectric coupling matrix
[ sI, sII ] = deal( [ ] );
for j = 1:EL_NN
sI = cat(2,sI,j:EL_NN);
sII = cat(2,sII,repmat(j,1,EL_NN-j+1));
end
[iKp,jKp] = deal(edofMatPZT(:,sI)',edofMatPZT(:,sII)');
Iar_p = sort([iKp(:),jKp(:)],2,'descend'); clear iKp jKp % Assembly indexing for piezoelectric dielectric stiffness matrix
%% ACTIVE & PASSIVE DOMAINS
Passive_el = ElNum([] ,:,:); Passive_el = Passive_el(:); % Definition of passive elements
Active_el = setdiff(1:nele,Passive_el); Active_el = Active_el(:);% Definition of active elements
%% DEFINITION OF BOUNDARY CONDITION
DE = ElNum(:,1,:); DE=DE(:); % Desired elements for left clamped side
DNN = LEFNODS; % Desired node numbers (elemental left nodes)
fixeddof = edofMat(DE,[3*DNN,3*DNN-1,3*DNN-2]); fixeddof = unique(fixeddof(:)); % Fix mechanical DOFs
freedofs = setdiff(1:ndof,fixeddof); freedofs=freedofs(:); lf = length(freedofs); % Free mechanical DOFs
%% DEFINITION OF ELECTRODES
DE = ElNum(1,:,:); DE = DE(:); % Desired elements for top electrode
DE2 = ElNum(nelz,:,:); DE2 = DE2(:); % Desired elements for bottom electrode
DE3 = ElNum(ceil(nelz/2),:,:); DE3 = DE3(:); % Desired elements for Middle electrode
TE = edofMatPZT(DE,TOPNODS); TE = unique(TE(:)); % Top electrode
BE = edofMatPZT(DE2,BOTNODS); BE = unique(BE(:)); % Bottom electrode
ME = edofMatPZT(DE3,BOTNODS); ME = unique(ME(:)); % Mid electrode
en = [TE;BE;ME]; en = unique(en(:)); % Equipotential nodes
pn = edofMatPZT(Passive_el,:); pn = unique(setdiff(pn(:),en(:))); % Nodes of passive elements
fn = setdiff(1:ndofPZT,[en;pn]); fn = unique(fn(:)); % FreeNodes
Uu = zeros(ndof,1);Up = zeros(ndofPZT,1);Adjoint = zeros(ndof,1); % Creation of null vectors
Up([TE;BE],1) = 1; % Actuation voltage
%% OUTPUT DISPLACEMENT DEFINITION
DE = ElNum(ceil(nelz/2),nelx,ceil(nely/2)); % Desired element
DNN = 6;% Desired node numbers
DMDOF = edofMat(DE,3*DNN-Dir); DMDOF=unique(sort(DMDOF(:))); % Desired mechanical degree of freedom
L = sparse(ndof,1); L(DMDOF,1) = -1;
%% SOLID & VOID DOMAINS
VOID = []; % Definition of void elements
SOLID = []; % Definition of solid elements
```

```
NVS = setdiff(1:nele,union(VOID(:),SOLID(:))); NVS = NVS(:); % Definition of non-void and non-solid elements
%% FILTERING
if ftBC == 'N', bcF = 'symmetric'; else, bcF = 0; end
prj = @(v,eta,beta) (tanh(beta*eta)+tanh(beta*(v(:)-eta)))./(tanh(beta*eta)+tanh(beta*(1-eta))); % projection
deta = @(v,eta,beta) - beta * csch( beta ) .* sech( beta * ( v( : ) - eta ) ).^2 .*sinh( v( : ) * beta ) .* sinh( ( 1 - v( : ) ) * beta );  % projection eta-derivative
dprj = @(v,eta,beta) beta*(1-tanh(beta*(v-eta)).^2)./(tanh(beta*eta)+tanh(beta*(1-eta)));% proj. x-derivative
cnt = @(v,vCnt,l) v+(l>=vCnt{1}).*(v<vCnt{2}).*(mod(l,vCnt{3})==0).*vCnt{4};
[dy,dz,dx] = meshgrid(-ceil(rmin)+1:ceil(rmin)-1,-ceil(rmin)+1:ceil(rmin)-1,-ceil(rmin)+1:ceil(rmin)-1 );
h = max( 0, rmin - sqrt( dx.^2 + dy.^2 + dz.^2 )); % Conv. kernel
Hp = imfilter( ones( nelz, nelx, nely ), h, bcF ); dHs = Hp; % Matrix of weights (filter)
%% INITIALIZE ITERATION
x = repmat(volfrac,nelz,nelx,nely); x(VOID) = 0; x(SOLID) = 1; xPhys = x; % Initial values for the densities
pol=repmat(0.5,[nelz,nelx,nely]); % Initial values for polarization
loop = 0;
Density_change = 1;
E0 = 1; Emin = 1e-9;
e0 = 1; eMin = 0;
eps0 = 1; epsMin = 1e-9;
as = []; % Initialize asymptotes
dv0 = ones(nelz,nelx,nely); % Volume sensitivity
penalratio_up = penalKup/penalKuu; penalratio_pp = penalKpp/penalKuu;  % Penalty ratios for continuation scheme
xold1 = [x(:);pol(:)]; % Vector of variables for previous iteration
xold2 = [x(:);pol(:)]; % Vector of variables for 2nd previous iteration
%% OPTIMIZATION ITERATIONS
while  Density_change > 1-6 && loop < Max_loop;    tic
loop = loop+1;
%% COMPUTE PHYSICAL DENSITY FIELD (AND ETA IF PROJECT.)
xTilde  = imfilter( reshape( x, nelz, nelx, nely ), h, bcF ) ./ Hp; xPhys(NVS) = xTilde(NVS); % Filtered field
if ft > 1 % Compute optimal eta* with Newton
f = ( mean( prj( xPhys, eta, beta ) ) - volfrac )  * (ft == 3); % Function (volume)
while abs( f ) > 1e-6 % Newton process for finding opt. eta
eta = eta - f / mean( deta( xPhys, eta, beta ) );
f = mean( prj( xPhys, eta, beta ) ) - volfrac;
end
dHs = Hp ./ reshape( dprj( xPhys, eta, beta ), nelz, nelx, nely ); % Sensitivity modification
xPhys = prj( xPhys, eta, beta ); % Projected (physical) field
end
%% FE-ANALYSIS
xPhys = reshape(xPhys,nelz,nelx,nely);
sK = ones(length(kuu_LT( : )),1).*(Emin+xPhys(:)'.^penalKuu*(E0-Emin));
sK(:,Active_el) = kuu_LT( : ).* sK(:,Active_el);
sK(:,Passive_el) = ks_LT( : ).* sK(:,Passive_el);
sKup = ones(length(kup(:)),1).*(eMin+xPhys(:)'.^penalKup*(e0-eMin).*((2*pol(:)-1)'.^penalPol));
sKup(:,Active_el) = kup(:).* sKup(:,Active_el);
sKup(:,Passive_el) = 0;
sKpp = ones(length(kpp_LT(:)),1)*(epsMin+xPhys(:)'.^penalKpp*(eps0-epsMin));
sKpp(:,Active_el)= kpp_LT(:).* sKpp(:,Active_el);
sKpp(:,Passive_el)= 0;
Kuu = sparse(Iar( :, 1 ),Iar( :, 2 ),sK(:)); Kuu = Kuu+Kuu'-diag(diag(Kuu)); % Global stifness matrix
for i=1: length(DMDOF);Kuu(DMDOF(i,1),DMDOF(i,1)) = Kuu(DMDOF(i,1),DMDOF(i,1))+Ks; end % Assembling the stifness of the modeled spring
Kup = sparse(Iar_up(:,1),Iar_up(:,2),sKup(:)); % Global piezoelectric coupling matrix
Kpp = sparse(Iar_p(:,1),Iar_p(:,2),sKpp(:)); Kpp = Kpp+Kpp'-diag(diag(Kpp)); % Global piezoelectric permittivity matrix
Ktot = [Kuu(freedofs,freedofs),Kup(freedofs,fn);Kup(freedofs,fn)',-gamma*Kpp(fn,fn)];
U = Ktot \ [-Kup(freedofs,en)*Up(en,:);gamma*Kpp(fn,en)*Up(en,:)]; % Response of the system
Uu(freedofs) = U(1:length(freedofs)); Up(fn) = U(length(freedofs)+1:end);
%% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
obj = full(sum(-L.*Uu)); % Objective Function
lambda = zeros (ndof,1);mu = zeros (ndofPZT,1);
ADJ =  Ktot \[L(freedofs,1);0*fn];
lambda(freedofs,:) = ADJ(1:lf,:); mu(fn,:) = ADJ(lf+1:end,:);
```

```
DCKuuE(Active_el) = sum((lambda(edofMat(Active_el,:))*kuu).*Uu(edofMat(Active_el,:)),2);
DCKuuE(Passive_el) = sum((lambda(edofMat(Passive_el,:))*ks).*Uu(edofMat(Passive_el,:)),2);
DCKupE(Active_el) = sum((lambda(edofMat(Active_el,:))*kup).*Up(edofMatPZT(Active_el,:)),2);
DCKupE(Passive_el) = 0;
DCKpuE(Active_el) = sum((Uu(edofMat(Active_el,:))*kup).*mu(edofMatPZT(Active_el,:)),2);
DCKpuE(Passive_el) = 0;
DCKppE(Active_el) = -gamma*sum((mu(edofMatPZT(Active_el,:))*kpp).*Up(edofMatPZT(Active_el,:)),2);
DCKppE(Passive_el) = 0;
DCKuu = reshape(DCKuuE,[nelz,nelx,nely]);
DCKup = reshape(DCKupE+DCKpuE,[nelz,nelx,nely]);
DCKpp = reshape(DCKppE,[nelz,nelx,nely]);
dc = penalKuu*(E0-Emin)*xPhys.^(penalKuu-1).*DCKuu+ penalKup*(E0-Emin)*((2*pol-1).^(penalPol)).*xPhys.^(penalKup-1).*DCKup+penalKpp*(eps0-epsMin)*xPhys.^(penalKpp-1).*DCKpp; % Sensitivity with respect to x
dp = 2*penalPol*((2*pol-1).^(penalPol-1)).*xPhys.^(penalKup).*DCKup; % Sensitivity with respect to p
dc = imfilter(reshape(dc,nelz,nelx,nely)./ dHs,h,bcF); % Filter objective sensitivity
dv = imfilter(reshape(dv0,nelz,nelx,nely)./ dHs,h,bcF); % Filter compliance sensitivity
%% UPDATING OPTIMIZATION VARIABLES
[Xupdate,as ,lmid ]= ocUpdate(loop , [x(:);pol(:)], [dc(:);dp(:)] ,[sum(xPhys(:))/(volfrac*nele) - 1],[dv(:)' / (volfrac*nele),0*pol(:)']' ,[move ,0.7 ,1.2] ,xold1 ,xold2 ,as , beta );
xnew = Xupdate(1:nele,1); xnew(VOID)=0; xnew(SOLID)=1; % Vector of updated density variables
Density_change = max(abs(xnew(:)-x(:)));
xold2 = xold1(:); xold1 = [x(:);pol(:)];
pol = reshape(Xupdate(nele+1:2*nele,1),nelz,nelx,nely); % Vector of updated polarization variables
x(NVS) = xnew (NVS);
%% CONTINIUATION SCHEME ON PENALIZATION FACTORS & BETA
[penalKuu ,beta] = deal(cnt(penalKuu ,penalCnt,loop),cnt(beta,betaCnt,loop));
penalKup=penalKuu*penalratio_up; penalKpp=penalKuu*penalratio_pp;
%% PLOT DENSITIES & POLARIZATION
fprintf(' It:%2.0i   Time:%3.2fs   Obj:%3.3e   Vol:%3.3f   ch:%3.3f\n ',loop,toc,obj,mean(xPhys(:)),Density_change);
Display(xPhys,pol,nelz,nelx,nely,Active_el,Passive_el,nele,ElNum)
end
%% PLOT DEFORMATION
Kuu = sparse(Iar( :, 1 ),Iar( :, 2 ),sK(:)); % Global stifness matrix
Kuu = Kuu+Kuu'-diag(diag(Kuu));
Ktot = [Kuu(freedofs,freedofs),Kup(freedofs,fn);Kup(freedofs,fn)',-gamma*Kpp(fn,fn)];
U =  Ktot \ [-Kup(freedofs,en)*Up(en,:);gamma*Kpp(fn,en)*Up(en,:)];  % Mechanical displacement
Uu(freedofs)=U(1:length(freedofs));
Deformation(Uu,xPhys,nelz,nelx,nely,edofMat,ElNum)
```

MATLAB 3D TOPOLOGY OPTIMIZATION CODE FOR PIEZOELECTRIC ENERGY HARVESTERS

```
% 3DTOPIEZO_ENERGY-HARVESTING // Abbas Homayouni-Amlashi et al. 2024
clc ;clear ;close all
%% GENERAL DEFINITIONS
La = 3e-2 ; Wa = 1.0e-2 ; Ha = 0.1e-2 ;  % Pieozoelectric geometrical dimensions (length, width, height) (m)
Lp = 3e-2 ; Wp = 1.0e-2 ; Hp = 0.1e-2 ;  % Passive material geometrical dimensions (length, width, height) (m)
nelx = 3*33 ; nely = 33 ; nelz = 4 ; % Number of elements in each direction
penalKuu = 3; penalKup = 6; penalKpp = 4 ; penalPol = 1 ; % Penalization factors
EL_T = 1; % Element type 1: trilinear, 2- quadratic
volfrac = 0.5; % Volume fraction
Max_loop = 120; % Maximum number of Iteration
pol_dir = 'z'; % Piezoelectric polarization direction
move = 0.2; % Optimization variable update move
ft = 2; % 1= Density filter, 2&3= projection with eta and beta as parameters
ftBC = 'N';
rmin = 1.5; % Filter radius
eta = 0.5; % Threshold
beta = 2; % Sharpness factor
penalCnt = {60,5,10,0.2}; % Continuation scheme on penalKuu {istart, maxPar, isteps, deltaPar}
betaCnt = {60,60,10,2}; % Continuation scheme on beta {istart, maxPar, isteps, deltaPar}
Dir=0; % Direction of force x=2; y=1; z=0;
omega = 150; % Excitation frequency (Hz)
wj = 0.1; % Objective function weigthing factor
Mass = 30; % Mass of attachement (gram)
%% MATERIAL PROPERTIES (PZT 4)
ro_p = 7500; % Density of piezoelectric material (kg/m^3)
C_p = [1.3900    0.7784    0.7428    0.0000    0.0000    0.0000
0.7784    1.3900    0.7428    0.0000    0.0000    0.0000
0.7428    0.7428    1.1541    0.0000    0.0000    0.0000
0.0000    0.0000    0.0000    0.2564    0.0000    0.0000
0.0000    0.0000    0.0000    0.0000    0.2564    0.0000
0.0000    0.0000    0.0000    0.0000    0.0000    0.3058]*1.0e+11; % Piezoelectric stiffness tensor
e   = [0.0000    0.0000    0.0000   0.0000  12.7179    0.0000
0.0000    0.0000    0.0000  12.7179   0.0000    0.0000
-5.2028   -5.2028   15.0804   0.0000   0.0000    0.0000]; % Piezoelectric coupling matrix
Ep  = [0.6746    0.0000    0.0000
0.0000    0.6746    0.0000
0.0000    0.0000    0.5867]*1.0e-08; % Piezoelectric permittivity matrix
ro_s = 2710; % Density of Passive material (kg/m^3)
EE = 70e9; % Young modulus of elasticity
NU = 0.30 ; % Poisson ratio
C_s = (EE/((1+NU)*(1-2*NU)))*[1-NU NU NU 0 0 0 ; NU 1-NU NU 0 0 0 ; NU NU 1- NU 0 0 0 ;
0 0 0 (1-2*NU)/2 0 0 ; 0 0 0 0 (1- 2*NU)/2 0 ; 0 0 0 0 0 (1- 2*NU)/2]; % Stiffness matrix for passive material
%% PREPARE FINITE ELEMENT ANALYSIS
[C_p_1,e_1,Ep_1] = Matrix_Rotation(C_p,e,Ep,pol_dir);
[kuu,kup,kpp,m_p,ndofPZT,EL_NN,TOPNODS,BOTNODS,FRNODS,BAKNODS,LEFNODS,RTNODS] = FEM(La,Wa,Ha,nelz,nelx,nely,C_p_1,e_1,Ep_1,ro_p,EL_T); % Pizeoelectric elemental matrices
[ks,~,~,m_s,~,~,~,~,~,~,~,~] = FEM(Lp,Wp,Hp,max(1,nelz-2),nelx,nely,C_s,zeros(3,6),zeros(3,3),ro_s,EL_T); % Passive material elemental matrices
k0 = max(abs(kuu(:)));beta0 = max(kpp(:));alpha0 = max(kup(:)); M0 = max(m_p(:));  % Normalization Factors
kuu = kuu/k0;ks = ks/k0;kup = kup/alpha0;kpp = kpp/beta0;gamma = (k0*beta0)/(alpha0^2); % Application of normalization I
m_p = m_p/M0; m_s = m_s/M0; omega = M0*(omega*2*pi)^2/k0; % Application of normalization II
kuu_LT = kuu(tril(true(size(kuu)))); % Vector of lower triangular matrix
kpp_LT = kpp(tril(true(size(kpp)))); % Vector of lower triangular element of piezoelectric dielectric stifness matrix
mp_LT = m_p(tril(true(size(m_p)))); % Vector of lower triangular element of piezoelectric mass matrix
ms_LT = m_s(tril(true(size(m_s)))); % Vector of lower triangular element of passive material mass matrix
ks_LT = ks(tril(true(size(ks)))); % Vector of lower triangular matrix
ndof = 3*ndofPZT; % Mechanical degrees of freedom
nele = nelx*nelz*nely; % Number of elements
ElNum = reshape(1:nele,nelz,nelx,nely); % Element indexing
% Building connectivity matrix
NNlinear=(nelz+1)*(nelx+1)*(nely+1);
edg1=reshape(1:NNlinear,nelz+1,nelx+1,nely+1);
```

```
edg2=reshape(NNlinear+1:NNlinear+(nely+1)*(nelx+1)*nelz,nelz,nelx+1,nely+1);
edg3=reshape(NNlinear+(nely+1)*(nelx+1)*nelz+1:NNlinear+(nely+1)*(nelx+1)*nelz+(nelz+1)*(nely+1)*nelx,nelz+1,nelx,nely+1);
edg4=reshape(NNlinear+(nely+1)*(nelx+1)*nelz+(nelz+1)*(nely+1)*nelx+1:...
NNlinear+(nely+1)*(nelx+1)*nelz+(nelz+1)*(nely+1)*nelx+nely*(nelz+1)*(nelx+1),nelz+1,nelx+1,nely);
n=0;
for i=1:nely
for j=1:nelx
for k = 1:nelz
n=n+1;
EDG1=edg1([k,k+1],[j,j+1],[i,i+1]);
EDG2=edg2(k,[j,j+1],[i,i+1]);
EDG3=edg3([k,k+1],j,[i,i+1]);
EDG4=edg4([k,k+1],[j,j+1],i);
ED(n,:)=[EDG1(:);EDG2(:);EDG3(:);EDG4(:)]';
end
end
end
EDM = ED(:,[2,4,3,1,6,8,7,5,14,10,13,9,20,19,17,18,16,12,15,11]);
edofMatPZT = EDM(:,1:EL_NN); % Electrical connectivitry matrix
edofMat(:,3:3:3*EL_NN)=3*edofMatPZT;edofMat(:,2:3:3*EL_NN)=3*edofMatPZT-1;edofMat(:,1:3:3*EL_NN)=3*edofMatPZT-2; % Mechanical connectivitry matrix
[sI,sII] = deal([]);
for j = 1:3*EL_NN
sI = cat(2,sI,j:3*EL_NN);
sII = cat(2,sII,repmat(j,1,3*EL_NN-j+1));
end
[iK,jK] = deal(edofMat(:,sI)',edofMat(:,sII)');
Iar = sort([iK(:),jK(:)],2,'descend'); clear iK jK % Assembly indexing (stiffness matrix)
[sI,sII] = deal([]);
for j = 1 : EL_NN
sI = cat(2,sI,1:3*EL_NN);
sII = cat(2,sII,repmat(j,1,3*EL_NN));
end
[iKup,jKup] = deal(edofMat(:,sI)',edofMatPZT(:,sII)');
Iar_up = [iKup(:),jKup(:)]; clear iKup jKup; % Assembly indexing for piezoelectric coupling matrix
[sI,sII] = deal([]);
for j = 1:EL_NN
sI = cat(2,sI,j:EL_NN);
sII = cat(2,sII,repmat(j,1,EL_NN-j+1));
end
[iKp,jKp] = deal(edofMatPZT(:,sI)',edofMatPZT(:,sII)');
Iar_p = sort([iKp(:),jKp(:)],2,'descend'); clear iKp jKp % Assembly indexing for piezoelectric dielectric stiffness matrix
%% ACTIVE & PASSIVE DOMAINS
Passive_el=ElNum([2:nelz-1],:,:); Passive_el=Passive_el(:); % Definition of passive elements
Active_el=setdiff(1:nele,Passive_el); Active_el=Active_el(:);% Definition of active elements
%% DEFINITION OF BOUNDARY CONDITION
DE = ElNum(:,1,:); DE=DE(:); % Desired element for left clamped side
DNN = LEFNODS; % Desired node numbers (elemental left nodes)
fixeddof = edofMat(DE,[3*DNN-1,3*DNN-2]); fixeddof = fixeddof(:); % Fix mechanical DOFs
freedofs = setdiff(1:ndof,fixeddof); lf = length(freedofs); % Free mechanical DOFs
%% DEFINITION OF ELECTRODES
PE1 = edofMatPZT(ElNum(1,:,:),TOPNODS); PE1 = PE1(:);
PE2 = edofMatPZT(ElNum(1,:,:),BOTNODS); PE2 = PE2(:);
PE3 = edofMatPZT(ElNum(nelz,:,:),TOPNODS); PE3 = PE3(:);
PE4 = edofMatPZT(ElNum(nelz,:,:),BOTNODS); PE4 = PE4(:);
en = [PE1;PE2;PE3;PE4]; en = unique(en(:)); % Equipotential nodes
pn = edofMatPZT(Passive_el,:); pn = unique(setdiff(pn(:),en(:))); % Nodes of passive elements
fn = setdiff(1:ndofPZT,[en;pn]); fn = unique(fn(:)); % FreeNodes
Nelec=2; % Number of potential electrodes
B = sparse(ndofPZT,Nelec); % Creation of null Bolean matrix
B(PE1',1)=1; B(PE4',2)=1;  % Creation of Bolean matrix
Up=zeros(ndofPZT,1);ADJ1 = zeros(ndof,1); ADJ2 = zeros(ndof,1); % Creation of null displacement vector
%% FORCE DEFINITION
nf = 1; % Number of forces
F = sparse(ndof,nf); % Definition of null vector for the force
DE = ElNum(:,1,:);DE=DE(:); % Desired element for appliation of force
DNN = LEFNODS; % Desired elemental node number
Fe=edofMat(DE,3*DNN-Dir); Fe = Fe(:); % Desired mechanical degrees of freedom
```

```
F(Fe,1) = +1; % Amplitude of the force
Ftot = [F(freedofs,:);zeros(length(fn),nf);zeros(Nelec,nf)];
%% DEFINITION OF ATTACHMENT MASS
sMass=zeros(nele,1);
sMass(ElNum([2:nelz-1],nelx,ceil(0.4*nely):ceil(0.6*nely)))=1; % Distribution of mass
le = Lp/nelx; we = Wp/nely; he = Ha; % Dimension of each element
ro_M = Mass*1e-3/(le*we*he)/length(find(sMass)); % Density of heavy elements
sMMass = (ro_M/ro_p)*mp_LT(:).*sMass';
sMMass = reshape(  sMMass, length( mp_LT(:) ) * nele, 1 );
M_Att = sparse(Iar( :, 1 ),Iar( :, 2 ),sMMass(:)); % Mass matrix containing only the attachement mass
%% SOLID & VOID DOMAINS
VOID = []; % Definition of void elements
SOLID = []; % Definition of solid elements
NVS = setdiff(1:nele,union(VOID(:),SOLID(:))); NVS = NVS(:); % Definition of non-void and non-solid elements
%% FILTER INITIALIZATION [Ferrari & Sigmund 2020]
if ftBC == 'N', bcF = 'symmetric'; else, bcF = 0; end
prj = @(v,eta,beta) (tanh(beta*eta)+tanh(beta*(v(:)-eta)))./(tanh(beta*eta)+tanh(beta*(1-eta))); % projection
deta = @(v,eta,beta) - beta * csch( beta ) .* sech( beta * ( v( : ) - eta ) ).^2 .*sinh( v( : ) * beta ) .* sinh( ( 1 - v( : ) ) * beta );  % projection eta-derivative
dprj = @(v,eta,beta) beta*(1-tanh(beta*(v-eta)).^2)./(tanh(beta*eta)+tanh(beta*(1-eta)));% proj. x-derivative
cnt = @(v,vCnt,l) v+(l>=vCnt{1}).*(v<vCnt{2}).*(mod(l,vCnt{3})==0).*vCnt{4};
[dy,dz,dx] = meshgrid(-ceil(rmin)+1:ceil(rmin)-1,-ceil(rmin)+1:ceil(rmin)-1,-ceil(rmin)+1:ceil(rmin)-1 );
h = max( 0, rmin - sqrt( dx.^2 + dy.^2 + dz.^2 )); % Conv. kernel
Hs = imfilter( ones( nelz, nelx, nely ), h, bcF ); dHs = Hs; % Matrix of weights (filter)
%% INITIALIZE ITERATION
x = repmat(volfrac,nelz,nelx,nely); x(VOID) = 0; x(SOLID) = 1; xPhys = x;  % Initial guess for the densities
pol = repmat(0.5,[nelz,nelx,nely]); % Initial values for polarization
loop = 0;
Density_change = 1;
E0 = 1; Emin = 1e-9;
e0 = 1; eMin = 1e-9;
eps0 = 1; epsMin = 1e-9;
as = []; % Initialize asymptotes
dv0 = ones(nelz,nelx,nely); % Volume sensitivity
penalratio_up = penalKup/penalKuu; penalratio_pp = penalKpp/penalKuu;  % Penalty ratios for continuation scheme
xold1 = [x(:);pol(:)]; % Vector of variables for previous iteration
xold2 = [x(:);pol(:)]; % Vector of variables for 2nd previous iteration
%% OPTIMIZATION ITERATIONS
while  loop < Max_loop; tic
loop = loop+1;
%% COMPUTE PHYSICAL DENSITY FIELD (AND ETA IF PROJECT.) [Ferrari & Sigmund 2020]
xTilde  = imfilter( reshape( x, nelz, nelx, nely ), h, bcF ) ./ Hs; xPhys(NVS) = xTilde(NVS);  % filtered field
if ft > 1    % Compute optimal eta* with Newton
f = ( mean( prj( xPhys, eta, beta ) ) - volfrac )  * (ft == 3);  % Function (volume)
while abs( f ) > 1e-6   % Newton process for finding opt. eta
eta = eta - f / mean( deta( xPhys, eta, beta ) );
f = mean( prj( xPhys, eta, beta ) ) - volfrac;
end
dHs = Hs ./ reshape( dprj( xPhys, eta, beta ), nelz, nelx, nely );   % Sensitivity modification
xPhys = prj( xPhys, eta, beta );     % Projected (physical) field
end
%% FE-ANALYSIS
xPhys = reshape(xPhys,nelz,nelx,nely);
sM = ones(length(mp_LT( : )),1).*xPhys(:)';
sM(:,Active_el)= mp_LT( : ).* sM(:,Active_el);
sM(:,Passive_el)= ms_LT( : ).* sM(:,Passive_el);
sM = reshape(  sM, length( mp_LT(:) ) * nele, 1 );
sKuu = ones(length(kuu_LT( : )),1).*(Emin+xPhys(:)'.^penalKuu*(E0-Emin));
sKuu(:,Active_el)= kuu_LT( : ).* sKuu(:,Active_el);
sKuu(:,Passive_el)= ks_LT( : ).* sKuu(:,Passive_el);
sKuu = reshape(sKuu, length(kuu_LT(:)) * nele, 1 );
sKup = ones(length(kup(:)),1)*(eMin+xPhys(:)'.^penalKup*(e0-eMin).*((2*pol(:)-1)'.^penalPol));
sKup(:,Active_el)=kup(:).* sKup(:,Active_el);
sKup(:,Passive_el)=zeros(size(kup(:))).*sKup(:,Passive_el);
sKpp = ones(length(kpp_LT(:)),1)*(epsMin+xPhys(:)'.^penalKpp*(eps0-epsMin));
sKpp(:,Active_el)= kpp_LT(:).* sKpp(:,Active_el);
sKpp(:,Passive_el)= zeros(size(kpp_LT(:))).*sKpp(:,Passive_el);
M= sparse(Iar( : , 1 ),Iar( : , 2 ),sM); % Global Mass matrix
Kuu = sparse(Iar( : , 1 ),Iar( : , 2 ),sKuu); KuuM = Kuu-(M)*omega; % Global stiffness matrix
Kuu = Kuu-(M+M_Att)*omega;Kuu = Kuu+Kuu'-diag(diag(Kuu)); % Global dynamic stiffness matrix
Kup = sparse( Iar_up( :, 1 ), Iar_up ( :, 2 ),sKup(:)); % Global piezoelectric coupling matrix
Kpp = sparse( Iar_p( :, 1 ), Iar_p ( :, 2 ),sKpp(:)); Kpp =Kpp+Kpp'-diag(diag(Kpp)); % Global piezoelectric permittivity matrix
Ktot = [Kuu(freedofs,freedofs),Kup(freedofs,fn),Kup(freedofs,en)*B(en,:);
Kup(freedofs,fn)',-gamma*Kpp(fn,fn),-gamma*Kpp(fn,en)*B(en,:);
B(en,:)'*Kup(freedofs,en)',-gamma*B(en,:)'*Kpp(fn,en)',-gamma*B(en,:)'*Kpp(en,en)*B(en,:)];
U =  ( Ktot \ Ftot); % Response of the system
Up(fn,:) = U(lf+1:lf+length(fn),:); Up(en,:) = B(en,:)*U(lf+length(fn)+1:end,:);
%% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
lambda1 = zeros (ndof,nf);lambda2 = zeros (ndof,nf);mu1 = zeros (ndofPZT,nf);mu2 = zeros (ndofPZT,nf);
ADJ1 =  Ktot\[-Kuu(freedofs,freedofs)*U(1:lf,:);zeros(length(fn),nf);zeros(Nelec,nf)]; % First adjoint vector
lambda1(freedofs,:) = ADJ1(1:lf,:); mu1(fn,:) = ADJ1(lf+1:lf+length(fn),:);mu1(en,:) = B(en,:)*ADJ1(lf+length(fn)+1:end,:);
ADJ2 =  Ktot\[zeros(lf,nf);-gamma*Kpp(fn,fn)*Up(fn,:)-gamma*Kpp(fn,en)*Up(en,:);-gamma*B(en,:)'*Kpp(en,en)*Up(en,:)-gamma*B(en,:)'*Kpp(en,fn)*Up(fn,:)]; % Second adjoint vector
lambda2(freedofs,:) = ADJ2(1:lf,:); mu2(fn,:) = ADJ2(lf+1:lf+length(fn),:);mu2(en,:) = B(en,:)*ADJ2(lf+length(fn)+1:end,:);
Uu_i = zeros(ndof,1); Wm = 0 ; We = 0 ; dc = zeros(nelz,nelx,nely) ; dp = zeros(nelz,nelx,nely);
for i = 1:nf
Uu_i(freedofs,1) = U(1:lf,i);Up_i = Up(:,i);
lambda1_i = lambda1(:,i); lambda2_i = lambda2(:,i);
mu1_i = mu1(:,i);mu2_i = mu2(:,i);
Wm = Wm+Uu_i'*KuuM*Uu_i; % Mechanical energy
We = We+Up_i'*Kpp*Up_i*gamma; % Electrical energy
dcKuuE(Active_el,:) = wj*((((1/2)*Uu_i(edofMat(Active_el,:)) + lambda1_i(edofMat(Active_el,:)))*kuu).*Uu_i(edofMat(Active_el,:)))-(1-wj)*((lambda2_i(edofMat(Active_el,:))*kuu).*Uu_i(edofMat(Active_el,:)));
dcKuuE(Passive_el,:) = wj*((((1/2)*Uu_i(edofMat(Passive_el,:)) + lambda1_i(edofMat(Passive_el,:)))*ks).*Uu_i(edofMat(Passive_el,:)))-(1-wj)*((lambda2_i(edofMat(Passive_el,:))*ks).*Uu_i(edofMat(Passive_el,:)));
dcKupE(Active_el,:) = wj*((lambda1_i(edofMat(Active_el,:))*kup).*Up_i(edofMatPZT(Active_el,:)) + ((Uu_i(edofMat(Active_el,:)))*kup).*mu1_i(edofMatPZT(Active_el,:)))-(1-wj)*((lambda2_i(edofMat(Active_el,:))*kup).*Up_i(edofMatPZT(Active_el,:)) + ((Uu_i(edofMat(Active_el,:)))*kup).*mu2_i(edofMatPZT(Active_el,:)));
dcKupE(Passive_el,:) = 0;
dcKppE(Active_el,:) = wj*((-mu1_i(edofMatPZT(Active_el,:))*kpp).*Up_i(edofMatPZT(Active_el,:)))-(1-wj)*((1/2)*(Up_i(edofMatPZT(Active_el,:))*kpp).*Up_i(edofMatPZT(Active_el,:)) - (mu2_i(edofMatPZT(Active_el,:))*kpp).*Up_i(edofMatPZT(Active_el,:)));
dcKppE(Passive_el,:) = 0;
dcME(Active_el,:) = wj*((((1/2)*Uu_i(edofMat(Active_el,:)) + lambda1_i(edofMat(Active_el,:)))*(-m_p*omega)).*Uu_i(edofMat(Active_el,:)))-(1-wj)*((lambda2_i(edofMat(Active_el,:))*(-m_p*omega)).*Uu_i(edofMat(Active_el,:)));
dcME(Passive_el,:) = wj*((((1/2)*Uu_i(edofMat(Passive_el,:)) + lambda1_i(edofMat(Passive_el,:)))*(-m_s*omega)).*Uu_i(edofMat(Passive_el,:)))-(1-wj)*((lambda2_i(edofMat(Passive_el,:))*(-m_s*omega)).*Uu_i(edofMat(Passive_el,:)));
dcKuu = reshape(full(sum(dcKuuE,2)),[nelz,nelx,nely]);
dcKup = reshape(full(sum(dcKupE,2)),[nelz,nelx,nely]);
dcKpp = gamma*reshape(full(sum(dcKppE,2)),[nelz,nelx,nely]);
dcM = reshape(full(sum(dcME,2)),[nelz,nelx,nely]);
dc = dc + penalKuu*(E0-Emin)*xPhys.^(penalKuu-1).*dcKuu+penalKup*(e0-eMin)*xPhys.^(penalKup-1).*dcKup.*((2*pol-1).^(penalPol))+penalKpp*(eps0-epsMin)*xPhys.^(penalKpp-1).*dcKpp+dcM; % Density variable sensitivity
dp = dp + (e0-eMin)*2*penalPol*((2*pol-1).^(penalPol-1)).*xPhys.^penalKup.*dcKup;   % Polarization variable sensitivity
end
c = wj*Wm-(1-wj)*We; % Objective function
dc = imfilter( reshape( dc, nelz, nelx, nely ) ./ dHs, h, bcF ); % Filter objective sensitivity
dv = imfilter( reshape( dv0, nelz, nelx, nely ) ./ dHs, h, bcF ); % Filter compliance sensitivity
%% UPDATING OPTIMIZATION VARIABLES (Ferrari & Sigmund 2020)
[Xupdate,as ,lmid ]= ocUpdate (loop , [x(:);pol(:)], [dc(:);dp(:)] ,[sum(xPhys(:))/(volfrac*nele) - 1],[dv(:)' / (volfrac*nele),0*pol(:)']' ,[move ,0.7 ,1.2] ,xold1 ,xold2 ,as , beta );
xnew = Xupdate(1:nele,1);xnew(VOID)=0;xnew(SOLID)=0;% Vector of updated density variables
```

```
Density_change = max(abs(xnew(:)-x(:)));
xold2 = xold1(:);xold1 = [x(:);pol(:)];
pol = reshape(Xupdate(nele+1:2*nele,1),nelz,nelx,nely); % Vector of updated polarization variables
x (NVS)= xnew(NVS);
%% CONTINIUATION SCHEME ON PENALIZATION FACTORS & BETA
[penalKuu ,beta] = deal(cnt(penalKuu ,penalCnt,loop),cnt(beta,betaCnt,loop));
penalKup=penalKuu*penalratio_up; penalKpp=penalKuu*penalratio_pp;
%% PRESENTATION OF RESULTS
fprintf(' It:%2.0i Time:%3.2fs Obj:%3.4e Wm.:%3.4e We.:%3.4e Vol:%3.3f ch:%3.3f\n ',loop,toc,c,Wm,
    We,mean(xPhys(:)),Density_change);
Display(xPhys,pol,nelz,nelx,nely,Active_el,Passive_el,nele,ElNum)
end
%% PLOT DEFORMATION (ELEMENTAL)
Uu = zeros(ndof,1);
Uu(freedofs,1) = U(1:lf,1);
Uu(edofMat(:,[3:3:24])) = Uu(edofMat(:,3:3:24))- Uu(edofMat(1,3));
Deformation(Uu,xPhys,nelz,nelx,nely,edofMat,ElNum) % Plot the deformation
%% Hints
% ** To run this code you have to add five addon functions :
% 1 Display, 2 ocUpdate, 3 FEM, 4 Matrix_Rotation, 5 Deformation
%
% ** With the addon functions, code can be compiled solely
%
% ||=====================================================================||
% || THIS CODE IS WRITTEN BY ABBAS HOMAYOUNIAMLASHI, OLE SIGMUND,       ||
% || THOMAS SCHLINQUER, MICKY RAKOTONDRABE, ABDENBI MOHANDOUSAID        ||
% || 2024                                                                ||
% ||=====================================================================||
```

ADD-ON FUNCTIONS : 1 - Display the results

```
%% PRESENTATION OF RESULTS (2D & 3D) // ABBAS HOMAYOUNI-AMLASHI 2024
function Display(xPhys,pol,nelz,nelx,nely,Active_el,Passive_el,nele,ElNum)
figure (1)
if nelz<=6;AX= subplot (4, nelz, [1:2*nelz]); cla();
else; AX=subplot (1, 2, 1); cla(); end
Xactive=zeros(nele,1);Xactive(Active_el,1)=xPhys(Active_el);
Xpassive=zeros(nele,1);Xpassive(Passive_el,1)=xPhys(Passive_el);
isovals_active = shiftdim( flipud(reshape( Xactive, nelz, nelx, nely )), 1);
isovals_active = smooth3( isovals_active, 'box', 1 );
patch(isosurface(isovals_active, .5),'FaceColor','m','EdgeColor','none');
patch(isocaps(isovals_active, .5),'FaceColor','m','EdgeColor','none');
isovals_Passive = shiftdim( flipud(reshape( Xpassive, nelz, nelx, nely )), 1);
isovals_Passive = smooth3( isovals_Passive, 'box', 1 );
patch(isosurface(isovals_Passive, .5),'FaceColor',[0 1 1],'EdgeColor','none');
patch(isocaps(isovals_Passive, .5),'FaceColor',[0 1 1],'EdgeColor','none');
isovals = shiftdim( reshape( ones(nele,1), nelz, nelx, nely ), 1);
isovals = smooth3( isovals, 'box', 1 );
patch(isosurface(isovals, .5),'FaceColor','none','EdgeColor','none');
patch(isocaps(isovals, .5),'FaceColor','none','EdgeColor','none');
set(AX,'XTick',[], 'YTick', [],'ZTick', []);
title('Density');view( [ 120, 30 ] ); axis equal tight; camlight; box on;AX.BoxStyle = 'full'; drawnow
if nelz>6; n=0;
for i=1:nelz; for j=1:nelx; for k=1:nely
if  xPhys (i,j,k)>0.9 && ismember(ElNum(i,j,k),Active_el)==1;
n=n+1; Coordinate(n,:)=0.01*[i,j,k];
COl(n,:) = [0.5-0.5*pol(i,j,k),0,0.5+0.5*pol(i,j,k)];
end
end; end; end
if exist('Coordinate')
AX2=subplot (1, 2, 2);
pcshow([Coordinate(:,2) Coordinate(:,3) Coordinate(:,1)],COl,'MarkerSize',120,'BackgroundColor',[1,1,1])
set(AX2,'XTick',[], 'YTick', [],'ZTick', []);title('Polarization');view( [ 30, 30 ] );axis equal tight;box on;AX2.BoxStyle = 'full';drawnow
end;end
if nelz<=6
for NL=2*nelz+1:3*nelz
XX(:,:)=xPhys(NL-2*nelz,:,:);
ax(NL)=subplot ( 4 , nelz , NL );
imagesc(1-XX(:,:)') ;colormap(ax(NL),gray) ;
set ( ax(NL) , 'XTick' , [ ] , 'YTick' , [ ] , 'XTicklabel' , [ ] ,...
'YTicklabel' , [ ] , 'xcolor' , 'w' , 'ycolor' , 'w')
xlabel ( sprintf ( 'Layer Number = %.0f' , NL-2*nelz ) , 'Color' , 'k')
title('Densities');
axis equal ; axis tight ; caxis([0 1]); drawnow ; hold on
end
for NL=3*nelz+1:4*nelz
XX(:,:)=(xPhys(NL-3*nelz,:,:).*(pol(NL-3*nelz,:,:).*2-1));
ax(NL)=subplot ( 4 , nelz , NL );
imagesc(XX(:,:)') ;colormap(ax(NL),jet) ;
set ( ax(NL) , 'XTick' , [ ] , 'YTick' , [ ] , 'XTicklabel' , [ ] ,...
'YTicklabel' , [ ] , 'xcolor' , 'w' , 'ycolor' , 'w')
xlabel ( sprintf ( 'Layer Number = %.0f' , NL-3*nelz ) , 'Color' , 'k')
title('Polarization');
axis equal ; axis tight ;caxis([-1 1]); drawnow ; hold on
end
else;end
end
```

ADD-ON FUNCTIONS : 2 - Updating algorithm

```
%% OCUpdate Algorithm (F. Ferrari et al. 2021)
function [x,as , lmid ]= ocUpdate (loop ,xT ,dg0 ,g1 ,dg1 ,ocPar ,xOld ,xOld1 ,as , beta )
[xU,xL] = deal ( min(xT+ ocPar (1) ,1) , max (xT-ocPar (1) ,0));
if loop <2.5 || beta > 4
as = xT +[ -0.5 ,0.5].*( xU -xL) ./( beta +1) ;
else
tmp = (xT - xOld ) .*( xOld - xOld1 );
gm = ones ( length (xT) ,1);
[gm(tmp >0) , gm(tmp <0) ] = deal ( ocPar (3) ,ocPar (2) );
```

```
as = xT + gm .* [-( xOld -as (: ,1)) ,(as (: ,2) -xOld )];
end
xL = max (0.9* as (: ,1) +0.1* xT ,xL); % adaptive lower bound
xU = min (0.9* as (: ,2) +0.1* xT ,xU); % adaptive upper bound
% ----- split (+) and (-) parts of the objective and constraint derivatives
p0_0 = (dg0 >0) .* dg0 ; q0_0 = (dg0 <0) .* dg0 ;
p1_0 = (dg1 >0) .* dg1 ; q1_0 = (dg1 <0) .* dg1 ;
[p0 ,q0] = deal ( p0_0 .*( as (: ,2) -xT).^2 , - q0_0 .*( xT -as (: ,1)) .^2) ;
[p1 ,q1] = deal ( p1_0 .*( as (: ,2) -xT).^2 , - q1_0 .*( xT -as (: ,1)) .^2) ;
% ------------------------ define the primal projection map and dual function
primalProj = @(lm) min (xU ,max (xL ,( sqrt (p0+lm*p1).* as (: ,1)+ sqrt (q0+lm*q1).* as (: ,2)) ...
./( sqrt (p0+lm*p1)+ sqrt (q0+lm*q1))));
psiDual = @(lm) g1-((as(:,2)-xT)'*p1_0-(xT-as(: ,1))'*q1_0)+sum(p1./(max(as(:,2)-primalProj(lm),1e-12))+q1./( max(primalProj(lm)-as (:,1),1e-12)));
% ------------------------- compute the Lagrange multiplier through bisection
lmUp = 1e6; x = xT; lmid = -1;
if psiDual ( 0 ) * psiDual ( lmUp ) < 0 % check if LM is within the interval
lmid = fzero ( psiDual , [ 0, lmUp ] );
x = primalProj ( lmid ); % update desing variables
elseif psiDual (0) < 0 % constraint cannot be active
lmid =0; x= primalProj ( lmid );
elseif psiDual ( lmUp ) > 0 % constraint cannot be fulfilled
lmid = lmUp ; x= primalProj ( lmid );
end
end
```

ADD-ON FUNCTIONS : 3 - Finite element matrices

```
%% FINITE ELEMENT MATRICES // ABBAS HOMAYOUNI-AMLASHI 2024
function [kuu,kup,kpp,m,ndofPZT,EL_NN,TOPNODS,BOTNODS,FRNODS,BAKNODS,LEFNODS,RTNODS] = FEM(L,W,H,nelz,nelx,nely,C,e,Ep,ro,EL_T)
le = L/nelx; we = W/nely; he=H; % Element geometry
if EL_T == 1 % Trilinear elements
g = 1/sqrt(3);
GP = [-g -g -g;g -g -g;g g -g;-g g -g;-g -g g;g -g g;g g g;-g g g]; % Gauss quadrature points
J = [ le/2, 0, 0; 0, we/2, 0; 0, 0, he/2]; % Jacobian matrix
detJ = (he*le*we)/8; % Determinant of Jacobian matrix
kuu = 0 ; kup = 0 ; kpp=0 ; m = 0; % Initial values for piezoelectric matrices
for ii=1:8
s=GP(ii,1);t=GP(ii,2);u=GP(ii,3); % s,t,u (natural coordinates)
N1 = (1/8)*(1-s)*(1+t)*(1-u);N2 = (1/8)*(1+s)*(1+t)*(1-u);N3 = (1/8)*(1+s)*(1+t)*(1+u);N4 = (1/8)*(1-s)*(1+t)*(1+u);
N5 = (1/8)*(1-s)*(1-t)*(1-u);N6 = (1/8)*(1+s)*(1-t)*(1-u);N7 = (1/8)*(1+s)*(1-t)*(1+u);N8 = (1/8)*(1-s)*(1-t)*(1+u);
N=[N1,0,0,N2,0,0,N3,0,0,N4,0,0,N5,0,0,N6,0,0,N7,0,0,N8,0,0;
0,N1,0,0,N2,0,0,N3,0,0,N4,0,0,N5,0,0,N6,0,0,N7,0,0,N8,0;
0,0,N1,0,0,N2,0,0,N3,0,0,N4,0,0,N5,0,0,N6,0,0,N7,0,0,N8]; % Matrix of interpolation functions
DN =[ ((t + 1)*(u - 1))/8, -((t + 1)*(u - 1))/8, ((t + 1)*(u + 1))/8, -((t + 1)*(u + 1))/8, -((t - 1)*(u - 1))/8, ((t - 1)*(u - 1))/8, -((t - 1)*(u + 1))/8, ((t - 1)*(u + 1))/8;
(s/8 - 1/8)*(u - 1), -(s/8 + 1/8)*(u - 1), (s/8 + 1/8)*(u + 1), -(s/8 - 1/8)*(u + 1), -(s/8 - 1/8)*(u - 1), (s/8 + 1/8)*(u - 1), -(s/8 + 1/8)*(u + 1), (s/8 - 1/8)*(u + 1);
(s/8 - 1/8)*(t + 1), -(s/8 + 1/8)*(t + 1), (s/8 + 1/8)*(t + 1), -(s/8 - 1/8)*(t + 1), -(s/8 - 1/8)*(t - 1), (s/8 + 1/8)*(t - 1), -(s/8 + 1/8)*(t - 1), (s/8 - 1/8)*(t - 1)]; % dN/d(s,t,u)
Bphi=J\DN; % Piezo Gradient interpolation matrix (Potential to electrical field matrix)
Bu(1,1:3:24)=Bphi(1,:);Bu(2,2:3:24)=Bphi(2,:);Bu(3,3:3:24)=Bphi(3,:);
Bu(6,1:3:24)=Bphi(2,:);Bu(6,2:3:24)=Bphi(1,:);
Bu(4,2:3:24)=Bphi(3,:);Bu(4,3:3:24)=Bphi(2,:);
Bu(5,1:3:24)=Bphi(3,:);Bu(5,3:3:24)=Bphi(1,:); % Strain-displacement matrix
kuu = kuu + transpose(Bu)*C*Bu*detJ; % Stiffness matrix
kup = kup + Bu'*e'*Bphi*detJ;  % Piezoelectric coupling matrix
kpp = kpp + Bphi'*Ep*Bphi*detJ; % Dielectric stiffness matrix
m = m+detJ*ro*(N'*N); % mass matrix
end
EL_NN = 8; % Elemental node numbers
ndofPZT = (nelx+1)*(nelz+1)*(nely+1); % Total electrical degrees of freedom
TOPNODS=[4,3,7,8]; BOTNODS=[1,2,5,6];
FRNODS=[5,6,7,8];  BAKNODS=[1,2,3,4];
LEFNODS=[1,4,5,8]; RTNODS=[2,3,6,7];
elseif EL_T == 2 % Quadratic elements
GPW = [-sqrt(3/5),5/9;0,8/9;sqrt(3/5),5/9]; % Gauss quadrature points and weights
x1=0; y1=we; z1=0; x2=le; y2=we; z2=0; x3=le; y3=we; z3=he; x4=0; y4=we; z4=he;
```

```matlab
x5=0; y5=0; z5=0; x6=le; y6=0; z6=0; x7=le; y7=0; z7=he; x8=0; y8=0; z8=he;
x9=(x1+x2)/2;y9=(y1+y2)/2;z9=(z1+z2)/2;
x10=(x2+x3)/2;y10=(y2+y3)/2;z10=(z2+z3)/2;
x11=(x3+x4)/2;y11=(y3+y4)/2;z11=(z3+z4)/2;
x12=(x1+x4)/2;y12=(y1+y4)/2;z12=(z1+z4)/2;
x13=(x2+x6)/2;y13=(y2+y6)/2;z13=(z2+z6)/2;
x14=(x3+x7)/2;y14=(y3+y7)/2;z14=(z3+z7)/2;
x15=(x4+x8)/2;y15=(y4+y8)/2;z15=(z4+z8)/2;
x16=(x1+x5)/2;y16=(y1+y5)/2;z16=(z1+z5)/2;
x17=(x5+x6)/2;y17=(y5+y6)/2;z17=(z5+z6)/2;
x18=(x6+x7)/2;y18=(y6+y7)/2;z18=(z6+z7)/2;
x19=(x8+x7)/2;y19=(y8+y7)/2;z19=(z8+z7)/2;
x20=(x8+x5)/2;y20=(y8+y5)/2;z20=(z8+z5)/2;
xs=le/2;xt=0;xu=0;ys=0;yt=we/2;yu=0;zs=0;zt=0;zu=he/2;
J = [xs ys zs; xt yt zt; xu yu zu]; % Jacobian matrix
detJ = xs*(yt*zu - zt*yu) - ys*(xt*zu - zt*xu) + zs*(xt*yu - yt*xu); % Determinant of Jacobian matrix
kuu = 0 ; kup = 0 ; kpp=0 ; m = 0; % Initial values for piezoelectric matrices
for i=1:3
s = GPW(i,1);
for j=1:3
t = GPW(j,1);
for k=1:3
u = GPW(k,1);
NV(1)=(1-s)*(1+t)*(1-u)*(-s+t-u-2)/8; NV(2)=(1+s)*(1+t)*(1-u)*(s+t-u-2)/8;
NV(3)=(1+s)*(1+t)*(1+u)*(s+t+u-2)/8;NV(4)=(1-s)*(1+t)*(1+u)*(-s+t+u-2)/8;
NV(5)=(1-s)*(1-t)*(1-u)*(-s-t-u-2)/8;NV(6)=(1+s)*(1-t)*(1-u)*(s-t-u-2)/8;
NV(7)=(1+s)*(1-t)*(1+u)*(s-t+u-2)/8;NV(8)=(1-s)*(1-t)*(1+u)*(-s-t+u-2)/8;
NV(9)=(1+t)*(1-u)*(1-s^2)/4;  NV(10)=(1+s)*(1+t)*(1-u^2)/4;
NV(11)=(1+t)*(1+u)*(1-s^2)/4; NV(12)=(1-s)*(1+t)*(1-u^2)/4;
NV(13)=(1+s)*(1-u)*(1-t^2)/4; NV(14)=(1+s)*(1+u)*(1-t^2)/4;
NV(15)=(1-s)*(1+u)*(1-t^2)/4;NV(16)=(1-s)*(1-u)*(1-t^2)/4;
NV(17)=(1-t)*(1-u)*(1-s^2)/4;NV(18)=(1+s)*(1-t)*(1-u^2)/4;
NV(19)=(1-t)*(1+u)*(1-s^2)/4;NV(20)=(1-s)*(1-t)*(1-u^2)/4;
N=zeros(3,60);N(1,1:3:60)=NV(:)';N(2,2:3:60)=NV(:)';N(3,3:3:60)=NV(:)';
DN=[-((t + 1)*(u - 1)*(2*s - t + u + 1))/8, -((t + 1)*(u - 1)*(2*s + t - u - 1))/8, ((t + 1)*(u + 1)*(2*s + t + u - 1))/8, ((t + 1)*(u + 1)*(2*s - t - u + 1))/8, ((t - 1)*(u - 1)*(2*s + t + u + 1))/8, -((t - 1)*(u - 1)*(t - 2*s + u + 1))/8, -((t - 1)*(u + 1)*(2*s - t + u - 1))/8, -((t - 1)*(u + 1)*(2*s + t - u + 1))/8, (s*(t + 1)*(u - 1))/2, -((u^2 - 1)*(t + 1))/4, -(s*(t + 1)*(u + 1))/2, ((u^2 - 1)*(t + 1))/4, ((t^2 - 1)*(u - 1))/4, -((t^2 - 1)*(u + 1))/4, ((t^2 - 1)*(u + 1))/4, -((t^2 - 1)*(u - 1))/4, -(s*(t - 1)*(u - 1))/2, ((u^2 - 1)*(t - 1))/4, (s*(t - 1)*(u + 1))/2, -((u^2 - 1)*(t - 1))/4;
-((s - 1)*(u - 1)*(s - 2*t + u + 1))/8, -((s + 1)*(u - 1)*(s + 2*t - u - 1))/8, ((s + 1)*(u + 1)*(s + 2*t + u - 1))/8, ((s - 1)*(u + 1)*(s - 2*t - u + 1))/8, ((s - 1)*(u - 1)*(s + 2*t + u + 1))/8, -((s + 1)*(u - 1)*(2*t - s + u + 1))/8, -((s + 1)*(u + 1)*(s - 2*t + u - 1))/8, -((s - 1)*(u + 1)*(s + 2*t - u + 1))/8, ((s^2 - 1)*(u - 1))/4, -((u^2 - 1)*(s + 1))/4, -((s^2 - 1)*(u + 1))/4, ((u^2 - 1)*(s - 1))/4, (t*(s + 1)*(u - 1))/2, -(t*(s + 1)*(u + 1))/2, (t*(s - 1)*(u + 1))/2, -(t*(s - 1)*(u - 1))/2, -((s^2 - 1)*(u - 1))/4, ((u^2 - 1)*(s + 1))/4, ((s^2 - 1)*(u + 1))/4, -((u^2 - 1)*(s - 1))/4;
-((s - 1)*(t + 1)*(s - t + 2*u + 1))/8, -((s + 1)*(t + 1)*(s + t - 2*u - 1))/8, ((s + 1)*(t + 1)*(s + t + 2*u - 1))/8, ((s - 1)*(t + 1)*(s - t - 2*u + 1))/8, ((s - 1)*(t - 1)*(s + t + 2*u + 1))/8, -((s + 1)*(t - 1)*(t - s + 2*u + 1))/8, -((s + 1)*(t - 1)*(s - t + 2*u - 1))/8, -((s - 1)*(t - 1)*(s + t - 2*u + 1))/8, ((s^2 - 1)*(t + 1))/4, -(u*(s + 1)*(t + 1))/2, -((s^2 - 1)*(t + 1))/4, (u*(s - 1)*(t + 1))/2, ((t^2 - 1)*(s + 1))/4, -((t^2 - 1)*(s + 1))/4, ((t^2 - 1)*(s - 1))/4, -((t^2 - 1)*(s - 1))/4, -((s^2 - 1)*(t - 1))/4, (u*(s + 1)*(t - 1))/2, ((s^2 - 1)*(t - 1))/4, -(u*(s - 1)*(t - 1))/2];
Bphi= double(J\DN);
B=zeros(6,60);
B(1,1:3:60)= Bphi(1,:);B(2,2:3:60)= Bphi(2,:);B(3,3:3:60)= Bphi(3,:);
B(6,1:3:60)= Bphi(2,:);B(6,2:3:60)= Bphi(1,:);
B(4,2:3:60)= Bphi(3,:);B(4,3:3:60)= Bphi(2,:);
B(5,1:3:60)= Bphi(3,:);B(5,3:3:60)= Bphi(1,:);
kuu = kuu + GPW(i,2)*GPW(j,2)*GPW(k,2)*transpose(B)*C*B*detJ;
kup = kup + GPW(i,2)*GPW(j,2)*GPW(k,2)*B'*e'*Bphi*detJ;
kpp = kpp+ GPW(i,2)*GPW(j,2)*GPW(k,2)*Bphi'*Ep*Bphi*detJ;
m = m + GPW(i,2)*GPW(j,2)*GPW(k,2)* detJ*ro*(N'*N);
end
end
end
EL_NN = 20; % Elemental node numbers
```

```
ndofPZT =((2*nelz+1)*(2*nelx+1)-nelx*nelz)*(nely+1)+(nelx+1)*(nelz+1)*nely; % Electrical degrees of freedom
TOPNODS=[4,3,7,8,11,14,15,19]; BOTNODS=[1,2,5,6,9,13,16,17];
FRNODS=[5,6,7,8,17,18,19,20];  BAKNODS=[1,2,3,4,9,10,11,12];
LEFNODS=[1,4,5,8,12,15,16,20]; RTNODS=[2,3,6,7,10,13,14,18];
end
end
```

ADD-ON FUNCTIONS : 4 - Rotation of coordinate system

```
%% TRANSFORMATION OF TENSOR CONSTANTS FOR ANISOTROPIC MATERIALS // ABBAS HOMAYOUNI-AMLASHI 2024
function [Cnew,enew,Epnew]=Matrix_Rotation(C,e,Ep,pol_dir)
if pol_dir == 'y' % For Polarization in the y direction
alpha= 0;beta=pi/2;gamma=0;
elseif pol_dir == 'z' % For Polarization in the z direction
alpha= 0;beta=0;gamma=0;
elseif pol_dir == 'x' % For Polarization in the x direction
alpha= 0;beta=pi/2;gamma=pi/2;
end
xi1=cos(gamma)*cos(alpha)-cos(beta)*sin(alpha)*sin(gamma);
xi2=-sin(gamma)*cos(alpha)-cos(beta)*sin(alpha)*cos(gamma);
xi3=sin(beta)*sin(alpha);
theta1=cos(gamma)*sin(alpha)+cos(beta)*cos(alpha)*sin(gamma);
theta2=-sin(gamma)*sin(alpha)+cos(beta)*cos(alpha)*cos(gamma);
theta3=-sin(beta)*cos(alpha);
psi1=sin(gamma)*sin(beta);psi2=cos(gamma)*sin(beta);psi3=cos(beta);
L=[xi1,theta1,psi1;xi2,theta2,psi2;xi3, theta3, psi3];
Z=[ xi1^2,theta1^2,psi1^2,2*theta1*psi1,2*psi1*xi1,2*xi1*theta1;
xi2^2,theta2^2,psi2^2,2*theta2*psi2,2*psi2*xi2,2*xi2*theta2;
xi3^2,theta3^2,psi3^2,2*theta3*psi3,2*psi3*xi3,2*xi3*theta3;
xi2*xi3,theta2*theta3,psi2*psi3,theta2*psi3+theta3*psi2,psi2*xi3+psi3*xi2,xi2*theta3+xi3*theta2;
xi3*xi1,theta3*theta1,psi3*psi1,theta1*psi3+theta3*psi1,psi1*xi3+psi3*xi1,xi1*theta3+xi3*theta1;
xi1*xi2,theta1*theta2,psi1*psi2,theta1*psi2+theta2*psi1,psi1*xi2+psi2*xi1,xi1*theta2+xi2*theta1];
Cnew=Z*C*Z'; % New stiffness matrix
enew=L*e*Z'; % New coupling matrix
Epnew=L*Ep*L'; % New permittivity matrix
end
```

ADD-ON FUNCTIONS : 5 - Plot deformation

```
%% PLOT DEFORMATION (ELEMENTAL) // ABBAS HOMAYOUNI-AMLASHI 2024
function Deformation(Uu,xPhys,nelz,nelx,nely,edofMat,ElNum)
figure (2); ax = gca;
AMP = 5/ max(abs(Uu(:))); % Amplification & normalization of deformation
view( [ 50, 20 ] ); % View angle
face = [ 1 2 3 4 ; 2 6 7 3 ; 4 3 7 8 ; 1 5 8 4 ; 1 2 6 5 ; 5 6 7 8 ];
xPhys=reshape(xPhys,nelz,nelx,nely);
for elz = 1:nelz
for elx = 1:nelx
for ely = 1:nely
if xPhys(elz,elx,ely)>0.5
Ue = AMP*Uu(edofMat(ElNum(elz,elx,ely),:));
ly = -(ely-nely)-1; lx = elx-1; lz = -(elz-nelz)-1;
xx_box = [lx lx+1 lx+1 lx lx lx+1 lx+1 lx]';
yy_box = [ly ly ly ly ly+1 ly+1 ly+1 ly+1]';
zz_box = [lz lz lz+1 lz+1 lz lz lz+1 lz+1]';
patch('Faces',face,'Vertices',[xx_box,yy_box,zz_box],'FaceColor','none')
xx = [Ue(1,1)+lx Ue(4,1)+lx+1 Ue(7,1)+lx+1 Ue(10,1)+lx Ue(13,1)+lx Ue(16,1)+lx+1 Ue(19,1)+lx+1 Ue(22,1)+lx]';
yy = [Ue(2,1)+ly+1 Ue(5,1)+ly+1 Ue(8,1)+ly+1 Ue(11,1)+ly+1 Ue(14,1)+ly Ue(17,1)+ly Ue(20,1)+ly Ue(23,1)+ly]';
zz = [Ue(3,1)+lz Ue(6,1)+lz Ue(9,1)+lz+1 Ue(12,1)+lz+1 Ue(15,1)+lz Ue(18,1)+lz Ue(21,1)+lz+1 Ue(24,1)+lz+1]';
Dis_R = max(abs(Uu(edofMat(ElNum(elz,elx,ely),:))))/max(abs(Uu(:))));
patch('Faces',face,'Vertices',[xx,yy,zz],'FaceColor',[Dis_R,0.4,1-Dis_R])
end
end
end
end
box on; ax.BoxStyle = 'full'; axis equal; set(gca,'XTick',[], 'YTick', [],'ZTick', []);axis off; drawnow; end
```

References

Alexandersen J, Andreasen CS (2020) A review of topology optimisation for fluid-based problems. Fluids 5(1):29

de Almeida BV, Cunha DC, Pavanello R (2019) Topology optimization of bimorph piezoelectric energy harvesters considering variable electrode location. Smart Materials and Structures

Andreasen CS, Elingaard MO, Aage N (2020) Level set topology and shape optimization by density methods using cut elements with length scale control. Structural and Multidisciplinary Optimization pp 1–23

Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O (2011) Efficient topology optimization in matlab using 88 lines of code. Structural and Multidisciplinary Optimization 43(1):1–16

Becker J, Fein O, Maess M, Gaul L (2006) Finite element-based analysis of shunted piezoelectric structures for vibration damping. Computers & structures 84(31-32):2340–2350

Bendsøe MP (1989) Optimal shape design as a material distribution problem. Structural optimization 1(4):193–202

Bendsøe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. Computer Methods in Applied Mechanics and Engineering

Bendsøe MP, Sigmund O (2003) Topology Optimization Theory, Methods and Applications. Springer Science & Business Media

Bower AF (2009) Applied mechanics of solids. CRC press

Cao Y, Huang H, Ding Y (2021) Isogeometric optimization of piezoelectric functionally graded material for energy harvester. Composite Structures 273:114261

Carbonari RC, Silva E, Nishiwaki S (2005) Design of multi-actuated piezoelectric mechanisms using topology optimization. In: Proceedings of the 6th world congress on structural and multidisciplinary optimization, Rio de Janeiro, Brazil, Citeseer

Chen S, Gonella S, Chen W, Liu WK (2010) A level set approach for optimal design of smart energy harvesters. Computer Methods in Applied Mechanics and Engineering 199(37-40):2532–2543

Christiansen RE, Sigmund O (2021) Inverse design in photonics by topology optimization: tutorial. JOSA B 38(2):496–509

Cook RD, et al. (2007) Concepts and applications of finite element analysis. John Wiley & Sons

Dbouk T (2017) A review about the engineering design of optimal heat transfer systems using topology optimization. Applied Thermal Engineering 112:841–854

Deaton JD, Grandhi RV (2014) A survey of structural and multidisciplinary continuum topology optimization: post 2000. Structural and Multidisciplinary Optimization 49(1):1–38

van Dijk NP, Maute K, Langelaar M, Van Keulen F (2013) Level-set methods for structural topology optimization: a review. Structural and Multidisciplinary Optimization 48(3):437–472

Donoso A, Bellido J (2009) Systematic design of distributed piezoelectric modal sensors/actuators for rectangular plates by optimizing the polarization profile. Structural and Multidisciplinary Optimization 38(4):347

Donoso A, Guest JK (2019) Topology optimization of piezo modal transducers considering electrode connectivity constraints. Computer Methods in Applied Mechanics and Engineering 356:101–115

Donoso A, Sigmund O (2016) Topology optimization of piezo modal transducers with null-polarity phases. Structural and Multidisciplinary Optimization 53(2):193–203

Du H, Lau GK, Lim MK, Qui J (2000) Topological optimization of mechanical amplifiers for piezoelectric actuators under dynamic motion. Smart materials and structures 9(6):788

Fahmy A, Adler E (1975) Computer-program description. transformation of tensor constants of anisotropic materials due to rotations of the coordinate axes. In: Proceedings of the Institution of Electrical Engineers, IET, vol 122, pp 591–592

Fattahi I, Mirdamadi HR (2019) A novel 3d skeletal frame topology for energy harvesting systems. Microelectronics Journal 83:6–17

Fattahi I, Mirdamadi HR (2020) Electro-vibration modeling and response of 3d skeletal frame configuration for energy harvesters. Extreme Mechanics Letters 36:100646

Ferrari F, Sigmund O (2020) A new generation 99 line matlab code for compliance topology optimization and its extension to 3d. Structural and Multidisciplinary Optimization 62(4):2211–2228

Ferrari F, Sigmund O, Guest JK (2021) Topology optimization with linearized buckling criteria in 250 lines of matlab. Structural and Multidisciplinary Optimization 63(6):3045–3066

Gałka A, Telega J, Wojnar R (1992) Homogenization and thermopiezoelectricity. Mechanics Research Communications 19(4):315–324

Gonçalves JF, De Leon DM, Perondi EA (2018) Simultaneous optimization of piezoelectric actuator topology and polarization. Structural and Multidisciplinary Optimization 58(3):1139–1154

Guo X, Zhang W, Zhong W (2014) Doing topology optimization explicitly and geometrically—a new moving morphable components based framework. Journal of Applied Mechanics 81(8)

Homayouni-Amlashi A, Ousaid AM, Rakotondrabe M (2019) Multi directional piezoelectric plate energy harvesters designed by topology optimization algorithm. IEEE Robotics and Automation Letters

Homayouni-Amlashi A, Mohand-Ousaid A, Rakotondrabe M (2020) Topology optimization of 2dof piezoelectric plate energy harvester under external inplane force. Journal of Micro-Bio Robotics pp 1–13

Homayouni-Amlashi A, Schlinquer T, Mohand-Ousaid A, Rakotondrabe M (2021) 2d topology optimization matlab codes for piezoelectric actuators and energy harvesters. Structural and Multidisciplinary Optimization 63(2):983–1014

Homayouni-Amlashi A, Rakotondrabe M, Mohand-Ousaid A (2023a) Structural design and frequency tuning of piezoelectric energy harvesters based on topology optimization. In: 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 5426–5432

Homayouni-Amlashi A, Schlinquer T, Kipkemoi P, Byiringiro JB, Rakotondrabe M, Gauthier M, Mohand-Ousaid A (2023b) Topology optimization of piezoelectric structures: micro-actuators and energy harvesters. In: 2023 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS), IEEE, pp 1–6

Homayouni-Amlashi A, Koebel L, Lefevre A, Mohand-Ousaid A, Bolopion A (2024a) Topology optimization of the electrodes in dielectrophoresis-based devices. Computers & Structures 301:107444

Homayouni-Amlashi A, Schlinquer T, Kipkemoi P, Byiringiro JB, Rakotondrabe M, Gauthier M, Mohand-Ousaid A (2024b) Topology optimization of micro piezoelectric actuators and energy harvesters at femto-st institute: summary and matlab code implementation. Journal of Micro and Bio Robotics 20(2):6

Hutton DV, Wu J (2004) Fundamentals of finite element analysis, vol 1. McGraw-hill New York

Kalimuldina G, Turdakyn N, Abay I, Medeubayev A, Nurpeissova A, Adair D, Bakenov Z (2020) A review of piezoelectric pvdf film by electrospinning and its applications. Sensors 20(18):5214

Kang Z, Tong L (2008) Integrated optimization of material layout and control voltage for piezoelectric laminated plates. Journal of Intelligent Material Systems and Structures 19(8):889–904

Kang Z, Wang X (2010) Topology optimization of bending actuators with multilayer piezoelectric material. Smart materials and structures 19(7):075018

Kang Z, Wang R, Tong L (2011) Combined optimization of bi-material structural layout and voltage distribution for in-plane piezoelectric actuation. Computer Methods in Applied Mechanics and Engineering 200(13-16):1467–1478

Kang Z, Wang X, Luo Z (2012) Topology optimization for static shape control of piezoelectric plates with penalization on intermediate actuation voltage. Journal of Mechanical Design 134(5)

Kim C, Shin JW (2013) Topology optimization of piezoelectric materials and application to the cantilever beams for vibration energy harvesting. International Journal of Precision Engineering and Manufacturing 14(11):1925–1931

Kim JE, Kim DS, Ma PS, Kim YY (2010) Multi-physics interpolation for the topology optimization of piezoelectric systems. Computer Methods in Applied Mechanics and Engineering 199(49-52):3153–3168

Kögl M, Silva EC (2005) Topology optimization of smart structures: design of piezoelectric plate and shell actuators. Smart materials and Structures 14(2):387

Lerch R (1990) Simulation of piezoelectric devices by two-and three-dimensional finite elements. IEEE transactions on ultrasonics, ferroelectrics, and frequency control 37(3):233–247

Li Y, Liu L, Yang S, Ren Z, Ma Y (2020) A multiobjective topology optimization methodology and its application to electromagnetic actuator designs. IEEE Transactions on Magnetics 56(2):1–4

Lin ZQ, Gea HC, Liu ST (2011) Design of piezoelectric energy harvesting devices subjected to broadband random vibrations by applying topology optimization. Acta Mechanica Sinica 27(5):730

Liu K, Tovar A (2014) An efficient 3d topology optimization code written in matlab. Structural and Multidisciplinary Optimization 50(6):1175–1196

Logan DL (2007) A first course in the finite element method, 4th edn. Thomson, Southbank

Malakooti MH, Sodano HA (2015) Piezoelectric energy harvesting through shear mode operation. Smart Materials and Structures 24(5):055005

Mattiat OE (2013) Ultrasonic transducer materials. Springer Science & Business Media

Meeker T (1996) Publication and proposed revision of ansi/ieee standard 176-1987. IEEE Transactions on Ultrasonics Ferroelectrics and Frequency Control 43(5):717–772

Molter A, Fonseca JSO, dos Santos Fernandez L (2016) Simultaneous topology optimization of structure and piezoelectric actuators distribution. Applied Mathematical Modelling 40(9-10):5576–5588

Moretti M, Silva EC (2019) Topology optimization of piezoelectric bi-material actuators with velocity feedback control. Frontiers of Mechanical Engineering 14(2):190–200

Motzki P, Seelecke S (2022) Encyclopedia of Smart Materials. Elsevier, Amsterdam, Netherlands

Nakasone P, Silva E (2010) Dynamic design of piezoelectric laminated sensors and actuators using topology optimization. Journal of Intelligent Material Systems and Structures 21(16):1627–1652

Noh JY, Yoon GH (2012) Topology optimization of piezoelectric energy harvesting devices considering static and harmonic dynamic loads. Advances in Engineering Software 53:45–60

Piefort V (2001) Finite element modelling of piezoelectric active structures. PhD thesis, Ph. D. thesis. Bruxelles, Belgium: Université Libre de Bruxelles, Department ...

Rozvany GI, Zhou M, Birker T (1992) Generalized shape optimization without homogenization. Structural optimization 4(3):250–252

Rupp CJ, Evgrafov A, Maute K, Dunn ML (2008) Optimal design of piezoelectric energy harvesters based on multilayer plates and shells. In: Smart Materials, Adaptive Structures and Intelligent Systems, vol 43314, pp 509–515

Rupp CJ, Evgrafov A, Maute K, Dunn ML (2009) Design of piezoelectric energy harvesting systems: a topology optimization approach based on multilayer plates and shells. Journal of Intelligent Material Systems and Structures 20(16):1923–1939

Salas R, Ramírez F, Montealegre-Rubio W, Silva E, Reddy J (2018) A topology optimization formulation for transient design of multi-entry laminated piezocomposite energy harvesting devices coupled with electrical circuit. International Journal for Numerical Methods in Engineering 113(8):1370–1410

Schlinquer T, Homayouni-Amlashi A, Rakotondrabe M, Ousaid AM (2020) Design of piezoelectric actuators by optimizing the electrodes topology. IEEE Robotics and Automation Letters 6(1):72–79

Sekhar BC, Dhanalakshmi B, Rao BS, Ramesh S, Prasad KV, Rao P, Rao BP (2021) Piezoelectricity and its applications. Multifunctional Ferroelectric Materials 11:13

Sigmund O (1997) On the design of compliant mechanisms using topology optimization. Journal of Structural Mechanics 25(4):493–524

Sigmund O (2001a) A 99 line topology optimization code written in matlab. Structural and multidisciplinary optimization 21(2):120–127

Sigmund O (2001b) Design of multiphysics actuators using topology optimization–part ii: Two-material structures. Computer methods in applied mechanics and engineering 190(49-50):6605–6627

Sigmund O, Maute K (2013) Topology optimization approaches. Structural and Multidisciplinary Optimization 48(6):1031–1055

Sigmund O, Torquato S, Aksay IA (1998) On the design of 1–3 piezocomposites using topology optimization. Journal of materials research 13(4):1038–1048

Silva ECN (1998) Design of piezocomposite materials and piezoelectric transducers using topology optimization. University of Michigan

Silva EN, Kikuchi N (1999) Design of piezocomposite materials and piezoelectric transducers using topology optimization–part iii. Archives of Computational Methods in Engineering 6(4):305

Silva EN, Fonseca JO, Kikuchi N (1997) Optimal design of piezoelectric microstructures. Computational mechanics 19(5):397–410

Silva EN, Nishiwaki S, Kikuchi N (1999) Design of piezocomposite materials and piezoelectric transducers using topology optimization—part ii. Archives of Computational Methods in Engineering 6:191–215

Svanberg K (1987) The method of moving asymptotes—a new method for structural optimization. International journal for numerical methods in engineering 24(2):359–373

Svanberg K (2007) Mma and gcmma-two methods for nonlinear optimization. vol 1:1–15

Tiersten HF (2013) Linear Piezoelectric Plate Vibrations: Elements of the Linear Theory of Piezoelectricity and the Vibrations Piezoelectric Plates. Springer

Townsend S, Grigg S, Picelli R, Featherston C, Kim HA (2019) Topology optimization of vibrational piezoelectric energy harvesters for structural health monitoring applications. Journal of Intelligent Material Systems and Structures 30(18-19):2894–2907

Venkayya V (1989) Optimality criteria: a basis for multidisciplinary design optimization. Computational mechanics 5(1):1–21

Wang C, Zhao Z, Zhou M, Sigmund O, Zhang XS (2021) A comprehensive review of educational articles on structural and multidisciplinary optimization. Structural and Multidisciplinary Optimization 64(5):2827–2880

Wang F, Lazarov BS, Sigmund O (2011) On projection methods, convergence and robust formulations in topology optimization. Structural and Multidisciplinary Optimization 43(6):767–784

Wang X, Lin Z, Ren Y (2017) Topology optimization of piezocomposite resonator for maximizing excitation strength and synthesizing desired eigenmodes. Acta Mechanica Solida Sinica 30(5):531–539

Wein F, Kaltenbacher M, Baensch E, Leugering G, Schury F (2009) Topology optimization of a piezoelectric-mechanical actuator with single-and multiple-frequency excitation. International Journal of Applied Electromagnetics and Mechanics 30(3-4):201–221

Wein F, Kaltenbacher M, Stingl M (2013) Topology optimization of a cantilevered piezoelectric energy harvester using stress norm constraints. Structural and Multidisciplinary Optimization 48(1):173–185

Weis R, Gaylord T (1985) Lithium niobate: Summary of physical properties and crystal structure. Applied Physics A 37(4):191–203

Xia L, Xia Q, Huang X, Xie YM (2018) Bi-directional evolutionary structural optimization on advanced structures and materials: a comprehensive review. Archives of Computational Methods in Engineering 25(2):437–478

Xie YM, Steven GP (1993) A simple evolutionary procedure for structural optimization. Computers & structures 49(5):885–896

Yang B, Cheng C, Wang X, Meng Z, Homayouni-Amlashi A (2022) Reliability-based topology optimization of piezoelectric smart structures with voltage uncertainty. Journal of Intelligent Material Systems and Structures p 1045389X211072197

Zhang W, Chen J, Zhu X, Zhou J, Xue D, Lei X, Guo X (2017) Explicit three dimensional topology optimization via moving morphable void (mmv) approach. Computer Methods in Applied Mechanics and Engineering 322:590–614

Zheng B, Chang CJ, Gea HC (2009) Topology optimization of energy harvesting devices using piezoelectric materials. Structural and Multidisciplinary Optimization 38(1):17–23

Zhu B, Zhang X, Zhang H, Liang J, Zang H, Li H, Wang R (2020) Design of compliant mechanisms using continuum topology optimization: A review. Mechanism and Machine Theory 143:103622