Technische Universität Bergakademie Freiberg

PERSONAL PROGRAMMING PROJECT

Implementation of Iso-geometric Analysis (IGA) for Piezoelectric Material

Vikas Diddige
64041

Supervised by
Dr. Sergii Kozinov

February 17, 2020

Abstract

Contents

1 Introduction

Among all the numerical methods Finite Element Methods (FEM) are more popularly used to find approximate solutions of partial differential equations. FEM approximates the Computer Aided Drawing (CAD) geometry by discretizing it into smaller geometries called elements. Such geometrical approximations may create numerical errors and seriously effect the accuracy of the solution. Isogeometric analysis (IGA) is a technique to generate geometry using CAD concept of Non Uniform Rational B-Splines (NURBS) and analyse using its basis functions. !The pioneers of this technique are Tom Hughes and his group at The University of Texas at Austin!.

1.1 Advantages of IGA over FEA

- The exact representation of the geometry for analysis rules out the possibility of geometrical approximations.
- A huge amount of time involved in finite element modelling can be avoided.

2 B-Splines

In this section a brief description of B-Splines is discussed since NURBS is an extended version of B-Splines. A B-Spline basis function is defined by its order and knot vectors. A B-spline basis function along with control points defines a B-Spline curve.

2.1 Order

For a point on a B-Spline curve the order of the basis function speaks about the number of nearby control points influence the given point. The degree p of the basis function is one less than the order of the curve.

2.2 Knot vector

A knot vector is an array with an ascending order of parameter values written as $\Xi = \{\xi_0, \xi_1, \xi_2 ....., \xi_{n+p}\}$. !The knot vector is a sequence of parameter values that determines where and how the control points affect the NURBS curve. The number of knots is always equal to the number of control points plus curve degree plus one (i.e. number of control points plus curve order). The knot vector divides the parametric space in the intervals mentioned before, usually referred to as knot spans. Each time the parameter value enters a new knot span, a new control point becomes active, while an old control point is discarded. It follows that the values in the knot vector should be in nondecreasing order, so (0, 0, 1, 2, 3, 3) is valid while (0, 0, 2, 1, 3, 3) is not.(WIKI)!

2.3 Control points

!The control points determine the shape of the curve.[8] Typically, each point of the curve is computed by taking a weighted sum of a number of control points. The weight of each point varies according to the governing parameter. For a curve of degree d, the weight of any control point is only nonzero in d+1 intervals of the parameter space. Within those intervals, the weight changes according to a polynomial function (basis functions) of degree d. At the boundaries of the intervals, the basis functions go smoothly to zero, the smoothness being determined by the degree of the polynomial.(WIKI)! The total number of control points is given by $n_{cp}(\xi)$ = total number of knots in $[\Xi] - (p+1)$. *****

2.4 B-Spline basis functions

For a given Knot vector Ξ, the B-spline basis function for polynomial degree ≥ 1 is defined by a recursive function

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

with

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & otherwise \end{cases} \tag{2}$$

2.4.1 Properties

1. $N_{i,0}(\xi)$ is a step wise function with a value 1 over the half open interval $\xi \in [\xi_i \leq \xi < \xi_{i+1})$ and zero on the rest.
2. Basis functions sum upto to unity $\sum_{i=0}^{n} N_{i,p}(\xi) = 1$
3. Basis functions are non-negative $N_{i,p}(\xi) \geq 0$ over the entire domain
4. ********** Write more***********

2.4.2 Derivatives

The first derivative of a B-Spline basis function with its variable ξ is given by

$$\frac{d}{d\xi} N_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{3}$$

Higher order derivatives are not necessary for IGA formulation.

2.5 B-Spline curves

A $pth - degree$ B-Spline curve with a set of control points P_i is given by

$$C(\xi) = \sum_{i=0}^{n} N_{i,p}(\xi) P_i \qquad \xi_0 \leq \xi \leq \xi_{n+p} \tag{4}$$

defined on the knot vector $\Xi = \{\xi_0, \xi_1, \xi_2 ....., \xi_{n+p}\}$

2.6 B-Spline surfaces

A B-Spline surface $S(\xi,\eta)$ is built by tensor product between B-Spline curves along each parametric direction (ξ,η). It requires knot vectors ($\Xi = \{\xi_0,\xi_1,\xi_2.....,\xi_{n+p}\}$,$H = \{\eta_0,\eta_1,\eta_2.....,\eta_{m+q}\}$) along each parametric direction and control net $P_{i,j}$

$$S(\xi,\eta) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(\xi)N_{j,q}(\eta)P_{i,j} \tag{5}$$

where p,q are the degrees of the B-Spline basis functions along ξ,η respectively and n,m are number of control points along ξ,η respectively. Eq. (6) can be compactly written as

$$S(\xi,\eta) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,j}^{p,q}(\xi,\eta)P_{i,j} \tag{6}$$

2.6.1 Derivatives

The partial derivatives of bivariate B-Spline basis functions w.r.t parametric co-ordinates is given as

$$\frac{\partial N_{i,j}^{p,q}(\xi,\eta)}{\partial\xi} = \frac{d}{d\xi}\Bigg(N_{i,p}(\xi)\Bigg)N_{j,q}(\eta) \qquad \frac{\partial N_{i,j}^{p,q}(\xi,\eta)}{\partial\eta} = \frac{d}{d\eta}\Bigg(N_{j,q}(\eta)\Bigg)N_{i,p}(\xi) \tag{7}$$

3 Non Uniform Rational B-Splines

NURBS are very often used in computer-aided design(CAD), manufacturing (CAM) and engineering (CAE) due to its flexibility to represent complex geometries. NURBS curves and surfaces are considered as the generalization of B-Spline and Bezier curves and surfaces. A NURBS basis function is defined by its order and knot vector.

3.1 NURBS basis functions

NURBS basis functions $R_{i,p}(\xi)$ are defined as

$$R_{i,p}(\xi) = \frac{N_{i,p}(\xi)w_i}{\sum_{i=0}^{n} N_{i,p}(\xi)w_i} \tag{8}$$

where $N_{i,p}(\xi)$ is the ith B-Spline basis function with order p and w_i denotes weight of the ith control point (P_i). ***** Explain weights with help of an example****. When $w_i = constant \quad \forall i$ the NURBS basis function reduces to B-Spline basis function.

3.1.1 Derivatives

The first derivative of a NURBS basis function with its variable ξ is given by

$$\frac{d}{d\xi}R_{i,p}(\xi) = \frac{N'_{i,p}(\xi)W(\xi) - N_{i,p}(\xi)W'(\xi)}{W^2(\xi)}w_i \tag{9}$$

where $N'_{i,p}(\xi) = \frac{d}{d\xi}N_{i,p}(\xi)$

and $W'(\xi) = \sum_{i=0}^{n} N'_{i,p}(\xi)w_i$

3.2 NURBS Curves

The p^{th} degree NURBS curve is given by

$$C(\xi) = \frac{\sum_{i=0}^{n} N_{i,p}(\xi) w_i P_i}{\sum_{i=0}^{n} N_{i,p}(\xi) w_i} \qquad \xi_0 \leq \xi \leq \xi_{n+p} \tag{10}$$

in short form

$$C(\xi) = \sum_{i=0}^{n} R_{i,p}(\xi) P_i \tag{11}$$

refer NURBS book and write more info

3.3 NURBS Surfaces and solids

NURBS Surfaces and solids are generated by the tensor product between NURBS curve basis functions.

1. NURBS Surfaces:
   A NURBS surface with degree p in ξ direction and degree q in η direction is defined as

$$S(\xi, \eta) = \sum_{i=0}^{n} \sum_{j=0}^{m} R_{i,j}^{p,q}(\xi, \eta) P_{i,j} \tag{12}$$

   where the bivariate NURBS basis functions are given by

$$R_{i,j}^{p,q}(\xi, \eta) = \frac{N_{i,p}(\xi) N_{j,q}(\eta) w_{i,j}}{\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(\xi) N_{j,q}(\eta) w_{i,j}} \tag{13}$$

2. NURBS Solids:
   A NURBS solid with degree p, q, k in ξ, η, ζ directions respectively is defined as

$$S(\xi, \eta, \zeta) = \sum_{i=0}^{n} \sum_{j=0}^{m} \sum_{k=0}^{l} R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) P_{i,j,k} \tag{14}$$

   where the $R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta)$ is given by

$$R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) = \frac{N_{i,p}(\xi) N_{j,q}(\eta) N_{k,l}(\zeta) w_{i,j,k}}{\sum_{i=0}^{n} \sum_{j=0}^{m} \sum_{k=0}^{l} N_{i,p}(\xi) N_{j,q}(\eta) N_{k,r}(\zeta) w_{i,j,k}} \tag{15}$$

   refer NURBS book and write more info

3.4 Derivatives of NURBS bivariate Basis Functions

The first partial derivatives of NURBS bivariate basis function are given by

$$\frac{\partial R_{i,j}^{p,q}}{\partial \xi} = \frac{N'_{i,p} N_{j,q} W - N_{i,p} N_{j,q} W'_{\xi}}{W^2} w_{i,j} \tag{16}$$

$$\frac{\partial R_{i,j}^{p,q}}{\partial \eta} = \frac{N_{i,p} N'_{j,q} W - N_{i,p} N_{j,q} W'_{\eta}}{W^2} w_{i,j} \tag{17}$$

where

$$W = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}N_{j,q}w_{i,j} \tag{18}$$

$$W_{\xi}^{'} = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}^{'}N_{j,q}w_{i,j} \tag{19}$$

$$W_{\eta}^{'} = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}N_{j,q}^{'}w_{i,j} \tag{20}$$

The first partial derivatives of the trivariate basis functions can be computed in a similar manner as bivariate basis functions using a chain rule.

4 Implementation Procedure for IGA

This section describes step-by-step implementation of the Isogeometric analysis. A modified FEM code structure can be used to implement IGA. Similar to FEM, IGA can be divided into pre-processing, processing and post-processing stages. The stages in an IGA analysis is shown with the help of an flow chart (2)

4.1 Pre-processing Stage of the Analysis

This subsection mainly deals with NURBS based geometry creation, types of assembly arrays needed to assemble discretized geometries and how to deal with homogeneous and non-homogeneous boundary conditions on boundary defining control points due to their higher (C^{p-1}) continuity unlike C^0 continuity of FEM nodes.

4.1.1 Geometry Creation

As mentioned before in the section 3, the construction of NURBS discretized geometry requires parametric details such as control points, knot vectors and the order of the NURBS curve. Commercial softwares like "Rhino" can be used to extract parametric details of the complex NURBS geometry.

4.1.2 Assembly arrays

Assembly arrays are required to assemble local discretized geometries to global geometry. For IGA as knot vector and control points defines the geometry two assembly arrays (1) Control point assembly array and (2) Knot vector connectivity array are required.

1. Control point assembly array:
   The degree of the NURBS curve determines the number of control points (n_{cp}^{e}) present in an IGA element. Considering a three-dimensional element (Ω^e)
   $n_{cp}^{e} = (p+1)(q+1)(r+1)$
   The details of the control points are stored row wise in assembly array. ***This can be illustrated with an example below

2. Knot vector connectivity array:
   The knot vector connectivity matrix is a row wise matrix with each row corresponds to respective element's knot spans. The columns corresponds to span ranges along ξ, η and ζ directions. ***This can be illustrated with an example below

4.1.3 Boundary Conditions

***Have to write

4.2 Processing Stage of the Analysis

In the processing stage of analysis it is required to compute global elemental stiffness matrix and global force vector and solve these for the solution field. To formulate these matrices it requires NURBS basis functions and their derivatives evaluation. Numerical integration scheme like Gauss-Legendre rule is employed to solve the volume and area integrals involved in forming the stiffness matrix and internal force vector. Numerical integration involves mapping elements from physical space to master space (which is also called unit domain). As NURBS basis functions are defined in parametric space as given in Eq. ()it requires an additional mapping from physical space to parametric space $(\Omega_e \rightarrow \widetilde{\Omega_e})$. Later parametric space can be mapped on to master space $((\widetilde{\Omega_e} \rightarrow \overline{\Omega_e}))$. This procedure is illustrated in Fig.(1).

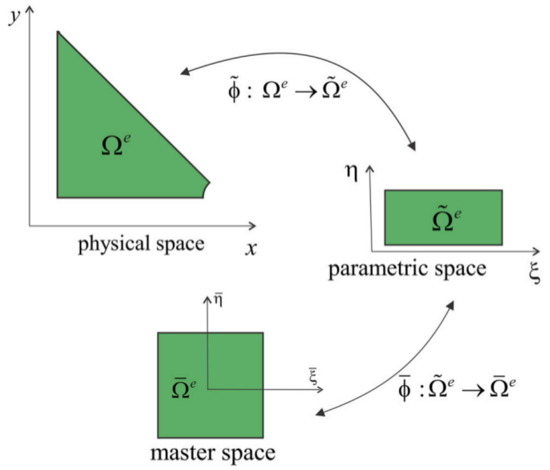


Figure 1:
Mapping IGA Physical element to Master element

1. Mapping from master space to parametric space:
   Consider a discretized IGA surface which is defined in parametric space $\widetilde{\Omega_e} = [\xi_i, \xi_{i+1}] \otimes [\eta_i, \eta_{i+1}]$, Refer Fig.(). The NURBS basis functions and their derivatives are evaluated at ξ, η of the element $\widetilde{\Omega_e}$. These ξ, η co-ordinate values are calculated by a linear mapping as shown below

$$\xi = \frac{1}{2}[(\xi_{i+1} - \xi_i)\overline{\xi} + (\xi_{i+1} + \xi_i)] \tag{21}$$

$$\eta = \frac{1}{2}[(\eta_{i+1} - \eta_i)\overline{\eta} + (\eta_{i+1} + \eta_i)] \tag{22}$$

where $\overline{\xi}, \overline{\eta}$ are the integration points defined in master space
$\mathbf{J}_2$ matrix is defined as

$$\mathbf{J}_2 = \frac{\partial \xi}{\partial \overline{\xi}} \frac{\partial \eta}{\partial \overline{\eta}} \tag{23}$$

Determinant of $\mathbf{J}_2$ matrix is required in numerical integration scheme for linear mapping

2. Mapping from physical space to parametric space:
The Jacobian matrix $\mathbf{J}_1$ used to map from physical space to parametric space ($\Omega_e \rightarrow \widetilde{\Omega_e}$) is computed as:

$$\mathbf{J}_1 = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} \tag{24}$$

The components of the $\mathbf{J}_1$ matrix are calculated using Eq. (32).

$$\frac{\partial x}{\partial \xi} = \sum_{k=1}^{n_{cp}^{e}} \frac{\partial \mathbf{R}_k}{\partial \xi} x_i \qquad \frac{\partial x}{\partial \eta} = \sum_{k=1}^{n_{cp}^{e}} \frac{\partial \mathbf{R}_k}{\partial \eta} x_i \tag{25}$$

$$\frac{\partial y}{\partial \xi} = \sum_{k=1}^{n_{cp}^{e}} \frac{\partial \mathbf{R}_k}{\partial \xi} y_i \qquad \frac{\partial y}{\partial \eta} = \sum_{k=1}^{n_{cp}^{e}} \frac{\partial \mathbf{R}_k}{\partial \eta} y_i \tag{26}$$

The two different mappings described above can be illustrated with an example considering $\mathbf{F}(x, y)$ integrated over the physical space Ω

$$\begin{aligned}
\int_{\Omega} \mathbf{F}(x, y) d\Omega &= \sum_{e=1}^{nel} \int_{\Omega_e} \mathbf{F}(x, y) d\Omega \\
&= \sum_{e=1}^{nel} \int_{\widetilde{\Omega_e}} \mathbf{F}(\xi, \eta) |\mathbf{J}_1| d\xi d\eta \\
&= \sum_{e=1}^{nel} \int_{\overline{\Omega_e}} \mathbf{F}(\overline{\xi}, \overline{\eta}) |\mathbf{J}_1| |\mathbf{J}_2| d\overline{\xi} d\overline{\eta} \\
&= \sum_{e=1}^{nel} \int_{-1}^{1} \int_{-1}^{1} \mathbf{F}(\overline{\xi}, \overline{\eta}) |\mathbf{J}_1| |\mathbf{J}_2| d\overline{\xi} d\overline{\eta} \\
&= \sum_{e=1}^{nel} \left[\sum_{i=1}^{n_{gp}^{e}} \mathbf{F}(\overline{\xi}_i, \overline{\eta}_i) gw_i |\mathbf{J}_1| |\mathbf{J}_2| d\overline{\xi} d\overline{\eta} \right]
\end{aligned}$$

where nel is the total number of elements and n_{gp}^{e}, gw_i denotes the number of Gauss points and their respective Gauss weights.
The formulated global stiffness matrix and force vector are solved using numerical scheme like Newton-Raphson method for the solution field.

4.3 Post-processing Stage of the Analysis

This section deals with visualization of the deformed geometry and how a displacement and stress fields are plotted.

1. Visualization of the deformed NURBS geometry:

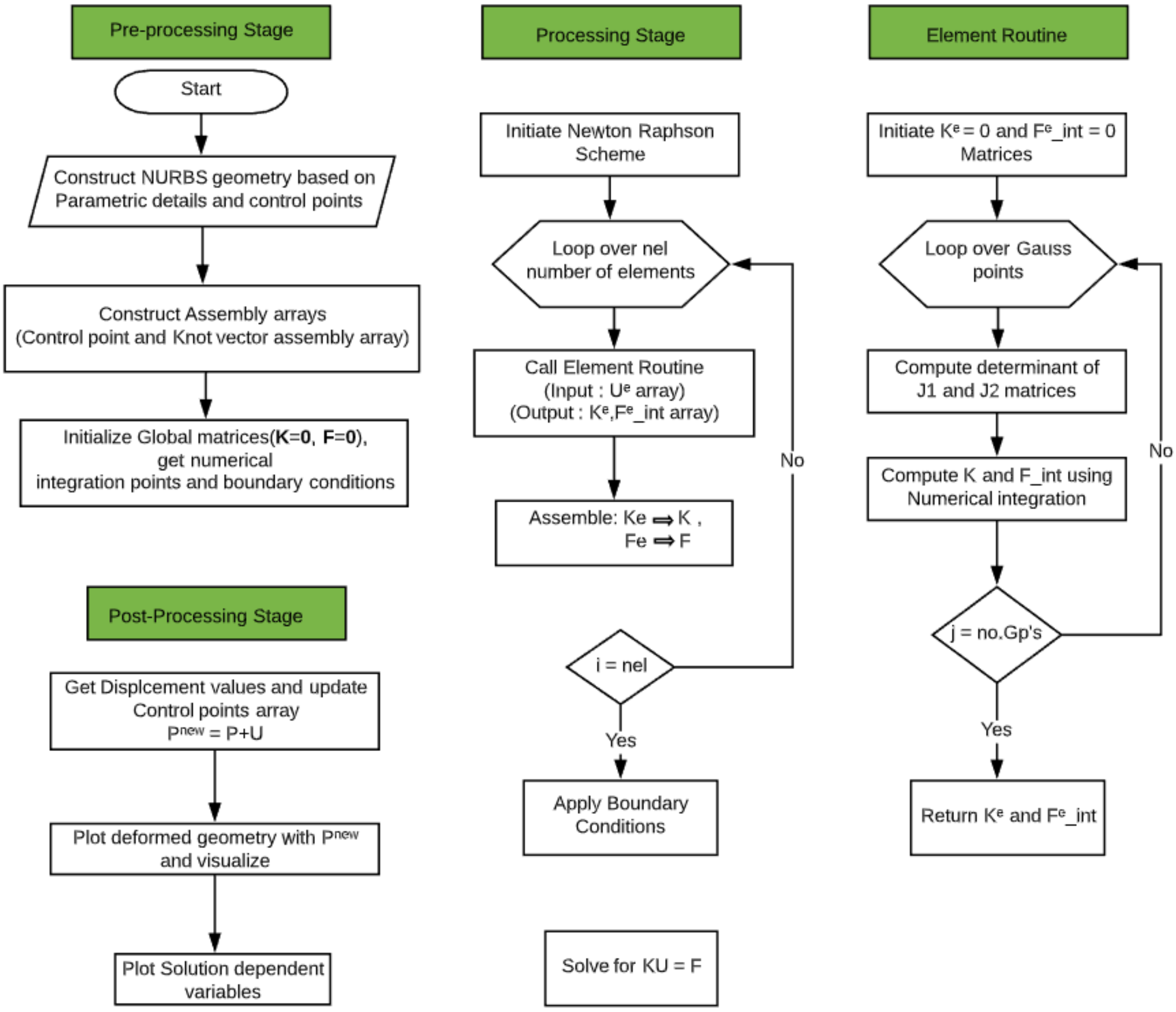
   Visualization of the deformed geometry will be done in the same manner as the visualization of the initial geometry. After determining the displacement field at control points they are added to the initial control points co-ordinates $[\mathbf{P}]$.

   $[\mathbf{P}_{new}] = [\mathbf{P}] + [\mathbf{U}]$

   $[\mathbf{P}^{new}]$ control points array is used to plot the deformed geometry.

2. Plotting of displacement and stress fields:

   *** Have to write



Pre-processing Stage
Start
Construct NURBS geometry based on Parametric details and control points
Construct Assembly arrays (Control point and Knot vector assembly array)
Initialize Global matrices(**K=0**, **F=0**), get numerical integration points and boundary conditions
Post-Processing Stage
Get Displcement values and update Control points array P^{new} = P+U
Plot deformed geometry with P^{new} and visualize
Plot Solution dependent variables
Processing Stage
Initiate Newton Raphson Scheme
Loop over nel number of elements
Call Element Routine (Input : U^e array) (Output : K^e,F^e_int array)
Assemble: Ke ⇒ K , Fe ⇒ F
No
i = nel
Yes
Apply Boundary Conditions
Solve for KU = F
Element Routine
Initiate K^e = 0 and F^e_int = 0 Matrices
Loop over Gauss points
Compute determinant of J1 and J2 matrices
Compute K and F_int using Numerical integration
No
j = no.Gp's
Yes
Return K^e and F^e_int

Figure 2:
Flow chart describing Pre-processing, Processing and Post-processing Stage of IGA

5 Mechanical Case

5.1 Governing Equations

The governing equation for mechanical deformation is based on conservation of linear momentum which can be written as

$$\sigma_{ij,i} + b_j = 0 \tag{27}$$

where σ_{ij} and b_i is the Cauchy stress tensor and body force. Due to the static nature of the analysis the inertial term is not included in the eq().
Stress and strain are related by following constitutive equation

$$\sigma_{ij} = c_{ijkl}\epsilon_{kl} \tag{28}$$

The infinitesimal strain theory is adopted for the analyis in which displacements of the material particles are considered to be very small compared to the dimentions of the body under loading. Strain in a small strain setting can be written as

$$\epsilon_{ij} = \frac{1}{2}[u_{i,j} + u_{j,i}] \tag{29}$$

where u_i are the displacements in the body

5.2 Weak Formulation

Consider a domain Ω with Γ_u as prescribed displacements and Γ_t as traction boundary conditions. The domain boundary can be represented as $\Gamma = \Gamma_u \cup \Gamma_t$ and $\Gamma_u \cap \Gamma_t = \Phi$
By using the principle of virtual work the eq() can be written as

$$\delta W = \int_\Omega (\sigma_{ij,i} + b_j)\delta u_j dV = 0 \tag{30}$$

with, $u = u_o$ on Γ_u (essential boundary condition) and $\sigma_{ij}n_j = \bar{t}_j$ on Γ_t (natural boundary condition)
where δu_j is the virtual displacement field, n_j is unit normal to the surface
Applying integration by parts to the stress term under integral and by making using of conservation of angular momentum ($\sigma_{ij} = \sigma_{ji}$) and Gauss divergence theorm (converting volume integral to surface integral) we approach at the following equation

$$\delta W = \int_\Omega \sigma_{ij}\epsilon_{ij} d\Omega - \left[\int_\Gamma \bar{t}_j \delta u_j d\Gamma_t + \int_\Omega b_j \delta u_j d\Omega\right] \tag{31}$$

as an additional requirement δu_j must be zero at essential boundary conditions (Γ_u) for a unique solution. ***Additional data regarding detailed explanation of steps to derive weak form can be included.

5.3 IGA Formulation

The advantage of IGA over FEM formulation lies in its basis functions incorporation and its ability to capture the exact geometry. While the FEM uses lagrangian basis functions, IGA uses NURBS basis functions which are used to generate the geometry itself. As discussed in the previous sections a multidimentional NURBS basis function is represented by $R^{p,q,r}_{i,j,k}(\xi,\eta,\zeta) \rightarrow R_i$. The isogeometric element is represented by basis function R_i and control points P_i as

$$\mathbf{x}^e = \sum_{i=1}^{n^e_{cp}} R_i P_i \tag{32}$$

By Galerkin approach, the displacements and virtual displacements are given by

$$\mathbf{u}^e = \sum_{i=1}^{n^e_{cp}} R_i \mathbf{u}_i \qquad \delta\mathbf{u}^e = \sum_{i=1}^{n^e_{cp}} R_i \delta\mathbf{u}_i \tag{33}$$

where $\mathbf{u}_i$ and $\delta\mathbf{u}_i$ are values at ith control point. The strain displacement matrix $\mathbf{B}$ is given by

$$\mathbf{B} = \begin{bmatrix} R_{1,x} & 0 & 0 & R_{2,x} & 0 & 0 & .... & R_{n^e_{cp},x} & 0 & 0 \\ 0 & R_{1,y} & 0 & 0 & R_{2,y} & 0 & .... & 0 & R_{n^e_{cp},y} & 0 \\ 0 & 0 & R_{1,z} & 0 & 0 & R_{2,z} & .... & 0 & 0 & R_{n^e_{cp},z} \\ R_{1,y} & R_{1,x} & 0 & R_{2,y} & R_{2,x} & 0 & .... & R_{n^e_{cp},y} & R_{n^e_{cp},x} & 0 \\ 0 & R_{1,z} & R_{1,y} & 0 & R_{2,z} & R_{2,y} & .... & 0 & R_{n^e_{cp},z} & R_{n^e_{cp},y} \\ R_{1,z} & 0 & R_{1,x} & R_{2,z} & 0 & R_{2,x} & .... & R_{n^e_{cp},z} & 0 & R_{n^e_{cp},x} \end{bmatrix} \tag{34}$$

By substituting Eqs, (33) and (34) in Eq.(31) the weak form in matrix terms can be written as

$$\sum_{e=1}^{nel} \left[\left(\int_{\Omega_e} \mathbf{B}^T \mathbf{C} \mathbf{B} d\Omega \right) \right] \mathbf{u} = \int_{\Gamma^e_t} \mathbf{R}^T.\mathbf{t} d\Gamma + \int_{\Omega^e_t} \mathbf{R}^T.\mathbf{f} d\Omega \tag{35}$$

where R is defined as
for the boundary Γ^e

$$\mathbf{R} = \begin{bmatrix} R_1(\xi,\eta) & 0 & R_2(\xi,\eta) & 0 & .... & R_{n^e_{cp}}(\xi,\eta) & 0 \\ 0 & R_1(\xi,\eta) & 0 & R_2(\xi,\eta) & .... & 0 & R_{n^e_{cp}}(\xi,\eta) \end{bmatrix} \tag{36}$$

for the domain Ω^e

$$\mathbf{R} = \begin{bmatrix} R_1(\xi,\eta,\zeta) & 0 & R_2(\xi,\eta,\zeta) & 0 & .... & R_{n^e_{cp}}(\xi,\eta,\zeta) & 0 \\ 0 & R_1(\xi,\eta,\zeta) & 0 & R_2(\xi,\eta,\zeta) & .... & 0 & R_{n^e_{cp}}(\xi,\eta,\zeta) \end{bmatrix} \tag{37}$$

Eq.(35) can be rewritten in a standard matrix form as

$$\sum_{e=1}^{nel} [\mathbf{K}^e \mathbf{U}^e = \mathbf{F}^e] \tag{38}$$

where $\mathbf{K}^e$ is isogeometric elements stiffness matrix, $\mathbf{U}^e$ is displacement vector and $\mathbf{F}^e$ force vector

6 Piezoelectric Case

6.1 Governing Equations for Piezoelectric Materials

The coupled electro-mechanical interactions are governed by conservation of momentum and Gauss's law as:

$$\sigma_{ij,j} + f_i = 0 \tag{39}$$

$$D_{i,i} - q = 0 \tag{40}$$

Where, f_i is body force, q is electrical charge, σ_{ij} is Cauchy stress tensor and D_i is electrical displacement vector. The constitutive equations for electromechanical coupling are defined as

$$\sigma_{ij} = C_{ijkl}\varepsilon_{kl} - e_{kij}E_k \tag{41}$$

$$D_i = e_{ikl}\varepsilon_{kl} + \kappa_{ik}E_k \tag{42}$$

Where, C_{ijkl} , e_{ikj} and ε_{ij} are elastic, piezoelectric and dielectric material constants respectively. The Cauchy stain tensor is defined as:

$$\varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}) \tag{43}$$

and electric field vector as:

$$E_i = -\phi_{,i} \tag{44}$$

6.2 Weak Formulation

Applying the principle of virtual work to the Eq.(39) and Eq. (40) we can write

$$\int_\Omega (\sigma_{ij,j} + f_i)\delta u_i d\Omega = 0 \tag{45}$$

$$\int_\Omega (D_{i,i} - q)\delta\phi d\Omega = 0 \tag{46}$$

with,
essential boundary conditions: $u = u_o$ on Γ_u and $\Phi = \Phi_0$ on Γ_Φ
natural boundary condition: $\sigma_{ij}n_j = \bar{t}_j$ on Γ_t and $D_in_i = q_0$ on Γ_q
where δu_i and $\delta\phi$ are virtual or arbitrary displacement and potential fields.
Integrating Eq. (45) and Eq. (46) by parts and later applying Guass divergence theorm and boundary conditions we approach at weak form

$$\int_\Omega \sigma_{ij}\delta\epsilon_{ij}d\Omega - \left[\int_\Gamma \bar{t}_i\delta u_i d\Gamma + \int_\Omega f_i\delta u_i d\Omega\right] = 0 \tag{47}$$

$$\int_\Omega D_i\delta E_i d\Omega - \left[\int_\Gamma Q\delta\phi d\Gamma + \int_\Omega q\delta\phi d\Omega\right] = 0 \tag{48}$$

6.3 IGA Formulation

By Galerkin approach, displacements, potentials and their virtual values are given by below equations

$$\mathbf{u}^e = \sum_{i=1}^{n_{cp}^e} R_i\mathbf{u}_i \qquad \delta\mathbf{u}^e = \sum_{i=1}^{n_{cp}^e} R_i\delta\mathbf{u}_i \tag{49}$$

$$\Phi^e = \sum_{i=1}^{n_{cp}^e} R_i\Phi_i \qquad \delta\Phi^e = \sum_{i=1}^{n_{cp}^e} R_i\delta\Phi_i \tag{50}$$

where $\mathbf{u}_i$, $\delta\mathbf{u}_i$, $\delta\Phi_i$ and Φ_i are values at ith control point.
By substituting Eqs, (49) and (50) in Eq.(47) and Eq.(48) the weak form in matrix notation can be written as

$$\begin{bmatrix} K_{MM} & K_{ME} \\ K_{EM} & K_{EE} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \phi \end{bmatrix} = \begin{bmatrix} F_M \\ F_E \end{bmatrix} \tag{51}$$

Where,

$$K_{MM} = \int_\Omega \mathbf{B}_u^T \mathbf{C} \mathbf{B}_u d\Omega \tag{52}$$

$$K_{ME} = \int_\Omega \mathbf{B}_u^T \mathbf{e} \mathbf{B}_e d\Omega \tag{53}$$

$$K_{EM} = \int_\Omega \mathbf{B}_e^T \mathbf{e}^T \mathbf{B}_u d\Omega \tag{54}$$

$$K_{EE} = -\int_\Omega \mathbf{B}_e^T \kappa \mathbf{B}_e d\Omega \tag{55}$$

$$F_M = \int_\Omega \mathbf{R}_u^T \mathbf{f} d\Omega + \int_\Gamma \mathbf{R}_u^T \mathbf{t} d\Gamma \tag{56}$$

$$F_E = \int_\Omega \mathbf{R}_e^T q d\Omega + \int_\Gamma \mathbf{R}_e^T Q d\Gamma \tag{57}$$

where R is defined as
for the boundary Γ

$$\mathbf{R}_u = \begin{bmatrix} R_1(\xi,\eta) & 0 & R_2(\xi,\eta) & 0 & .... & R_{n_{cp}^e}(\xi,\eta) & 0 \\ 0 & R_1(\xi,\eta) & 0 & R_2(\xi,\eta) & .... & 0 & R_{n_{cp}^e}(\xi,\eta) \end{bmatrix} \tag{58}$$

$$\mathbf{R}_e = \begin{bmatrix} R_1(\xi,\eta) & R_2(\xi,\eta) & .... & R_{n_{cp}^e}(\xi,\eta) \end{bmatrix} \tag{59}$$

for the domain Ω

$$\mathbf{R}_u = \begin{bmatrix} R_1(\xi,\eta,\zeta) & 0 & R_2(\xi,\eta,\zeta) & 0 & .... & R_{n_{cp}^e}(\xi,\eta,\zeta) & 0 \\ 0 & R_1(\xi,\eta,\zeta) & 0 & R_2(\xi,\eta,\zeta) & .... & 0 & R_{n_{cp}^e}(\xi,\eta,\zeta) \end{bmatrix} \tag{60}$$

$$\mathbf{R}_e = \begin{bmatrix} R_1(\xi,\eta,\zeta) & R_2(\xi,\eta,\zeta) & .... & R_{n_{cp}^e}(\xi,\eta,\zeta) \end{bmatrix} \tag{61}$$

B matix is given as

$$\mathbf{B}_u = \begin{bmatrix} R_{1,x} & 0 & 0 & R_{2,x} & 0 & 0 & .... & R_{n_{cp}^e,x} & 0 & 0 \\ 0 & R_{1,y} & 0 & 0 & R_{2,y} & 0 & .... & 0 & R_{n_{cp}^e,y} & 0 \\ 0 & 0 & R_{1,z} & 0 & 0 & R_{2,z} & .... & 0 & 0 & R_{n_{cp}^e,z} \\ R_{1,y} & R_{1,x} & 0 & R_{2,y} & R_{2,x} & 0 & .... & R_{n_{cp}^e,y} & R_{n_{cp}^e,x} & 0 \\ 0 & R_{1,z} & R_{1,y} & 0 & R_{2,z} & R_{2,y} & .... & 0 & R_{n_{cp}^e,z} & R_{n_{cp}^e,y} \\ R_{1,z} & 0 & R_{1,x} & R_{2,z} & 0 & R_{2,x} & .... & R_{n_{cp}^e,z} & 0 & R_{n_{cp}^e,x} \end{bmatrix} \tag{62}$$

$$\mathbf{B}_e = \begin{bmatrix} R_{1,x} & R_{2,x} & .... & R_{n_{cp}^e,x} \\ R_{1,y} & R_{2,y} & .... & R_{n_{cp}^e,y} \\ R_{1,z} & R_{2,z} & .... & R_{n_{cp}^e,z} \end{bmatrix} \tag{63}$$

$$\varepsilon = \mathbf{B}_u.\mathbf{u} \tag{64}$$

$$\mathbf{E} = -\mathbf{B}_e.\Phi \tag{65}$$

Eq. (51) can be solved using numerical methods like Newton Raphson scheme for displacements and potential solution field.

7 Modelling and Results

7.1 2D Plate with linear elastic loading

7.1.1 Problem description

A 2D plate is subjected to mechanical loading as shown in Figure(**??**).The properties of the material are, Youngs modulus E = 210GPa and Poisson's ratio $\nu = 0.3$. The movement of bottom edge is fixed in y direction and left edge in x direction. A displacement of 0.1 mm on right edge and 0.2 mm on top edge in x and y directions are given respectively.

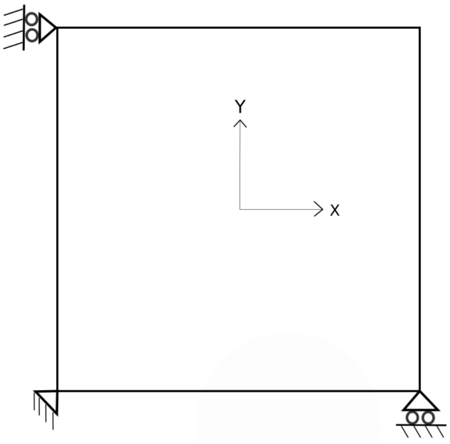


Figure 3: 2D Plate

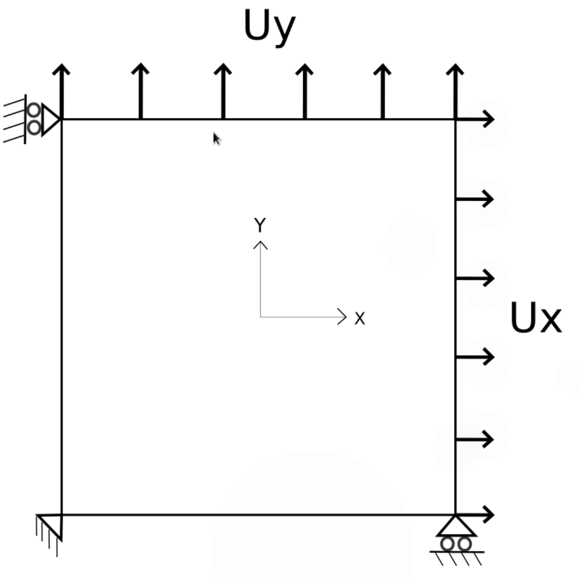


Figure 4: 2D Plate with loading

7.1.2 Parametric details for the plate with single element

The 2nd order NURBS curve is used in both ξ and η directions.

1. Physical details for the geometry:
   L = 10 # Length of the plate in mm
   H = 10 # Height of the plate in mm

2. Parametric details of the geometry:
   Ξ = [0,0,1,1] # Knot vector in xi direction

H = [0,0,1,1] # Knot vector in eta direction

Degree of the curve
p=1 # Degree of the curve in ξ direction
q=1 # Degree of the curve in η direction

Number of control points in each direction
n^{ξ}_{cp} = len(Ξ) - (p+1) #No.of control points in ξ direction (4-(1+1) = 2)
n^{η}_{cp} = len(H) - (q+1) #No.of control points in η direction (4-(1+1) = 2)

3. Total number of control points for the geometry
$n_{cp} = n^{\xi}_{cp} * n^{\eta}_{cp}$ = 2*2 = 4
The control points are given by

i	$P_{i,0}$	$P_{i,1}$
0	$(0, 0, 0, 1)$	$(0, 10, 0, 1)$
1	$(10, 0, 0, 1)$	$(10, 10, 0, 1)$

with fourth value in the paranthesis being weights of respective control points. As this is the case of single element there is no need for the control point assembly array and knot vector connectivity matrix.

7.1.3 Results and discussions

Abaqus plane strain full integration piezoelectric element (**CPE4**) is used for analysis. The below figures shows the values of displacements (U), electrical potentials (EPOT) and reaction forces (RF) for both abaqus and IGA element.
Figure(5) and Figure(6) shows the displacement (U1) values of the single CPE4 element and single IGA element at 100 % loading in x direction respectively.

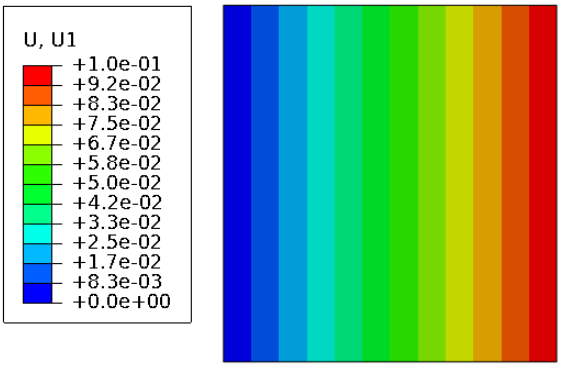


Figure 5: CPE4 Element:U1

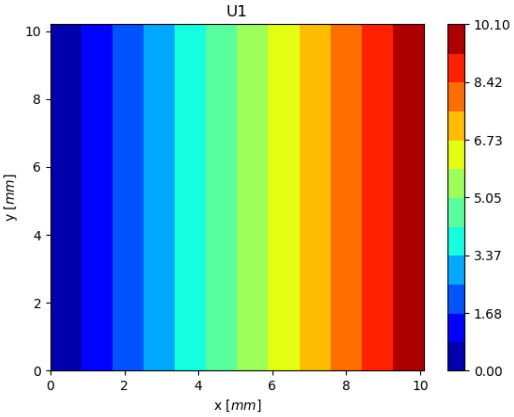


Figure 6: IGA Element:U1

Figure(7) and Figure(8) shows the displacement (U2) values of the single CPE4 element and single IGA element at 100 % loading in y direction respectively.

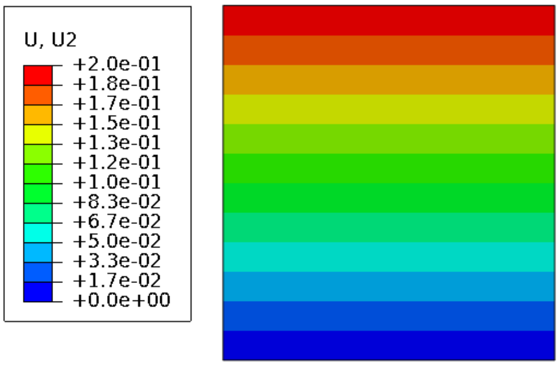


Figure 7: CPE4 Element:U2

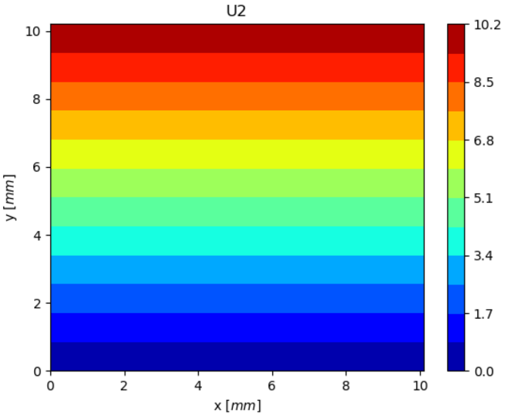


Figure 8: IGA Element:U2

Figure(9) and Figure(10) shows the Reaction force values (RF1) of the single CPE4 element and single IGA element at 100 % loading in x direction respectively.

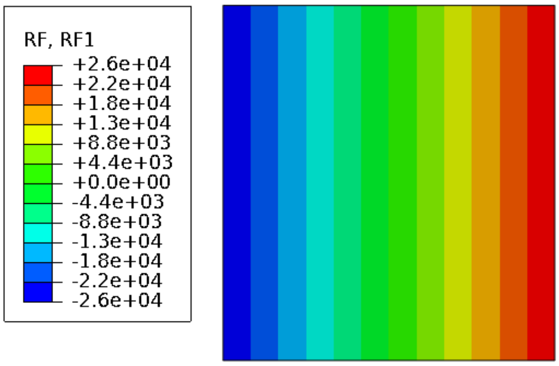


Figure 9: CPE4 Element:RF1

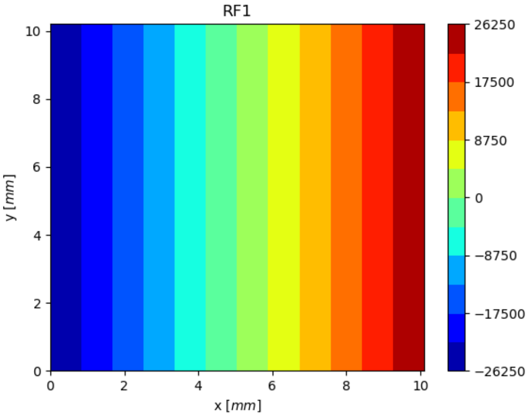


Figure 10: IGA Element:RF1

Figure(11) and Figure(12) shows the Reaction force values (RF2) of the single CPE4 element and single IGA element at 100 % loading in y direction respectively.

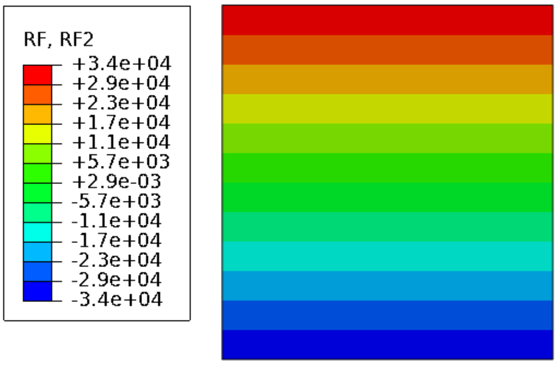


Figure 11: CPE4 Element:RF2

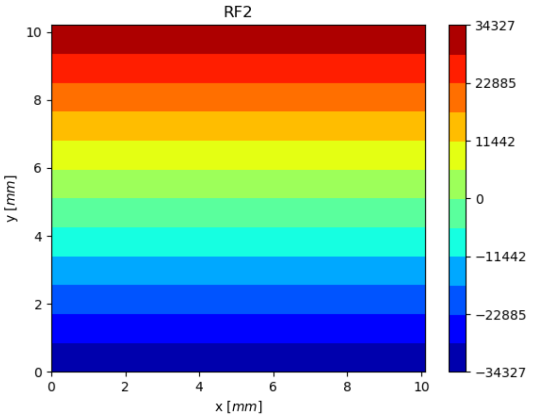


Figure 12: IGA Element:RF2

7.1.4 Conclusion

As shown in figures above the values generated by IGA code are inline with the results of Abaqus element. So it can be concluded that IGA code written works fine for 2D one element case.

7.2 2D Piezoelectric plate

7.2.1 Problem description

A 2D piezoelectric plate subjected to mechanical displacements and electrical loading is considered as shown in Figure(13) . The material used is PZT4 and material properties are as follows

$$C = \begin{bmatrix} 139000 & 74280 & 77840 & 0 & 0 & 0 \\ 47280 & 115400 & 74280 & 0 & 0 & 0 \\ 77840 & 74280 & 139000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 25640 & 0 & 0 \\ 0 & 0 & 0 & 0 & 25640 & 0 \\ 0 & 0 & 0 & 0 & 0 & 25640 \end{bmatrix} MPa$$

$$e = \begin{bmatrix} 0 & -5.20710E-6 & 0 \\ 0 & 15.08E-6 & 0 \\ 0 & -5.207E-6 & 0 \\ 12.710E-6 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 12.710E-6 \end{bmatrix} C/mm`2$$

$$\varepsilon = \begin{bmatrix} 6.752E-12 & 0 & 0 \\ 0 & 5.872E-12 & 0 \\ 0 & 0 & 6.752E-12 \end{bmatrix} C/(Vmm)$$

where
C is Elastic constants, e is Piezoelectric constants and ε being dielectric constants. The material is polarized in Y-direction.

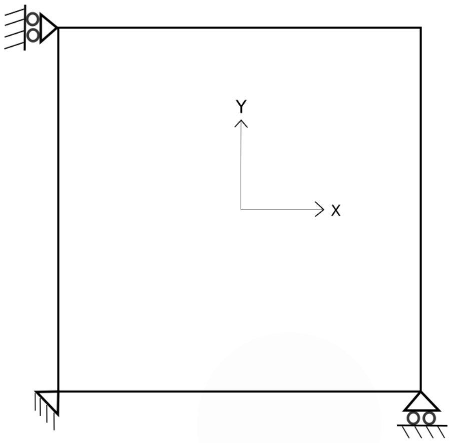


Figure 13: 2D Piezoelectric Plate

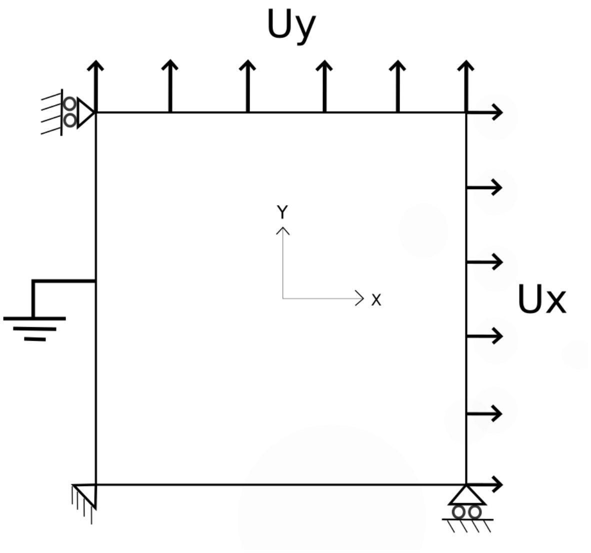


Figure 14: 2D Piezoelectric Plate with loading

The movement of the bottom edge and left edge of 2D piezoelectric plate is fixed in y direction and x direction respectively as shown in figure(14). The left edge is grounded (Electric potential $\Phi = 0$)and a displacement load of 0.1 mm and 0.2 mm is applied on the right edge and top edge respectively. The results for single element case and multiple elements is discussed in the below sections.
The results generated by IGA code is compared with inbuilt Abaqus piezoelectric element.

7.2.2 Results and discussions

Abaqus plane strain full integration piezoelectric element (**CPE4E**) is used for analysis. The below figures shows the values of displacements (U), electrical potentials (EPOT) and reaction forces (RF) for both abaqus and IGA element.
Figure(15) and Figure(16) shows the displacement (U1) values of the single CPE4E element and single IGA element at 100 % loading in x direction respectively.

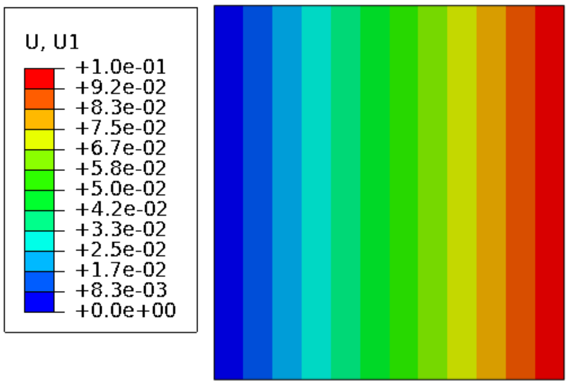


Figure 15: CPE4E Element:U1

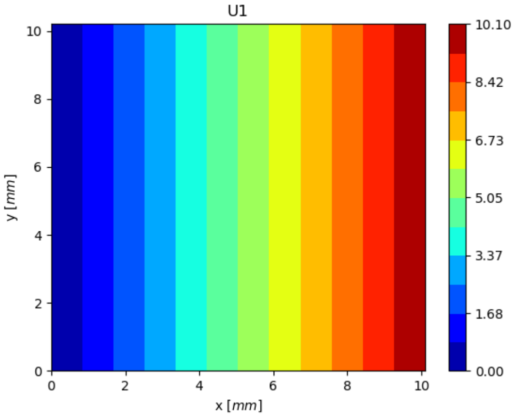


Figure 16: IGA Piezoelectric Element:U1

Figure(17) and Figure(18) shows the displacement (U2) values of the single CPE4E element and single IGA element at 100 % loading in y direction respectively.

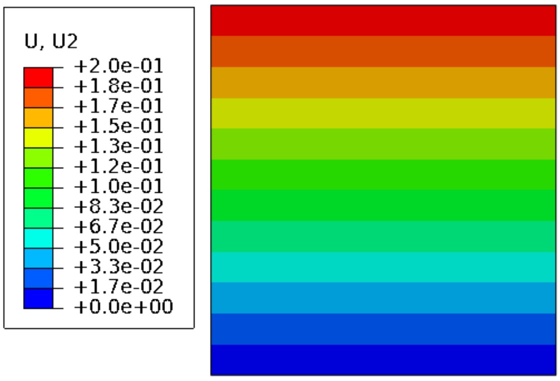


Figure 17: CPE4E Element:U2

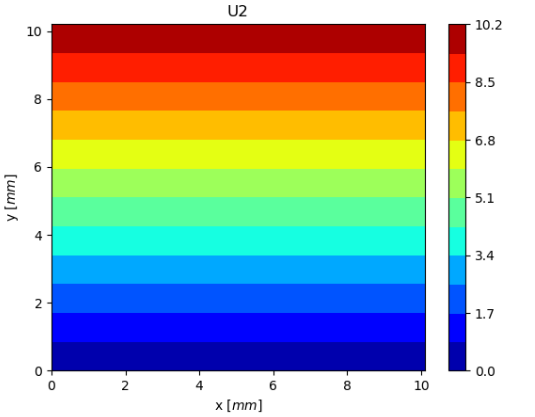


Figure 18: IGA Piezoelectric Element:U2

Figure(19) and Figure(20) shows the Reaction force values (RF1) of the single CPE4E element and single IGA element at 100 % loading in x direction respectively.

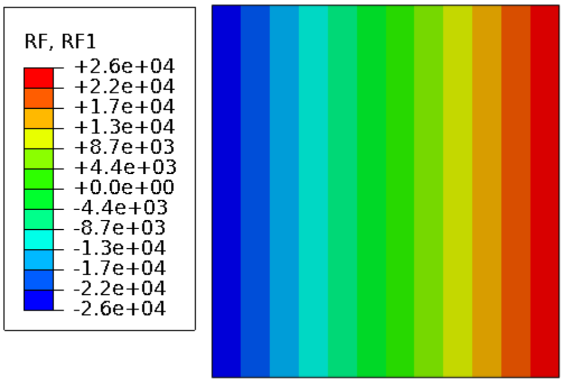


Figure 19: CPE4E Element:RF1

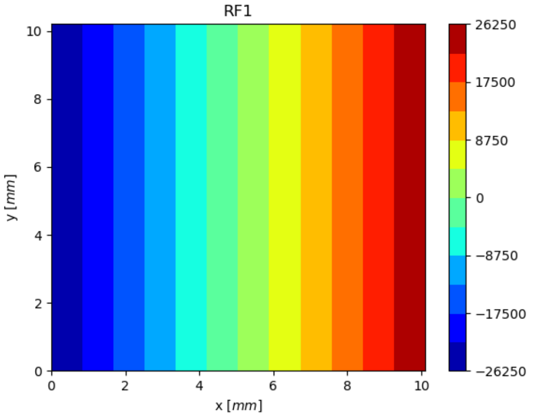


Figure 20: IGA Piezoelectric Element:RF1

Figure(21) and Figure(22) shows the Reaction force values (RF2) of the single CPE4E element and single IGA element at 100 % loading in y direction respectively.

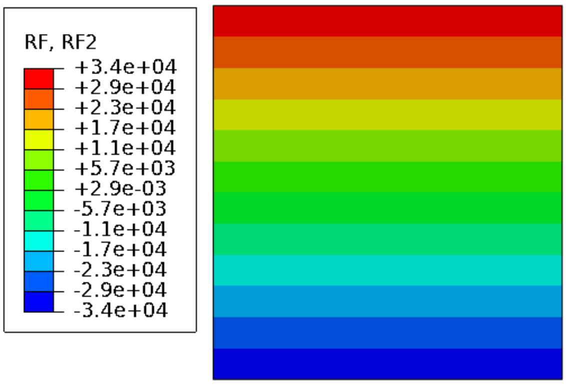


Figure 21: CPE4E Element:RF2

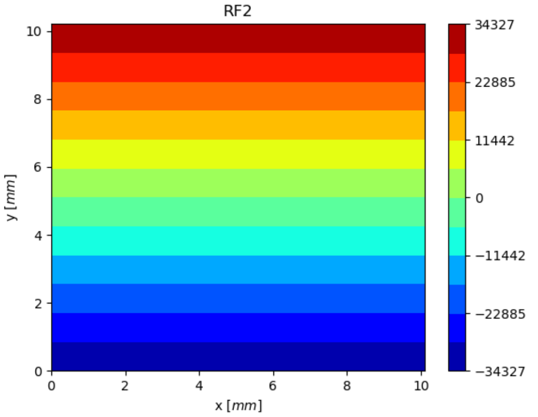


Figure 22: IGA Piezoelectric Element:RF2

Figure(23) and Figure(24) shows the Electrical potential values (EPOT) of the single CPE4E element and single IGA element at 100 % loading respectively.

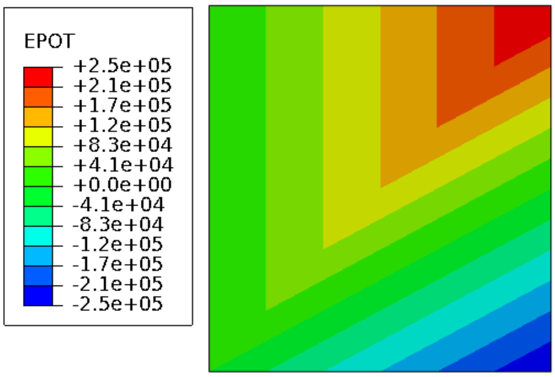


Figure 23: CPE4E Element:EPOT

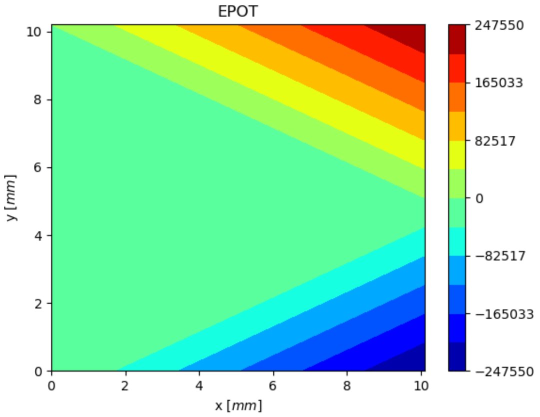


Figure 24: IGA Piezoelectric Element:EPOT

7.2.3 Conclusion

As shown in figures above the values generated by IGA code are inline with the results of Abaqus element. So it can be concluded that IGA code written works fine for 2D one element case.

7.2.4 Parametric details for the plate with 3 elements in x direction and 2 elements in y direction

The 2nd order NURBS curve is used in both ξ and η directions.

1. Physical details for the geometry:
   L = 10 # Length of the plate in mm
   H = 10 # Height of the plate in mm

2. Parametric details of the geometry:
   Ξ = [0,0,1,2,2] # Knot vector in xi direction
   H = [0,0,1,2,3,3] # Knot vector in eta direction

   Degree of the curve
   p=1 # Degree of the curve in ξ direction
   q=1 # Degree of the curve in η direction

   Number of control points in each direction
   n_{cp}^{ξ} = len(Ξ) - (p+1) #No.of control points in ξ direction (5-(1+1) = 3)
   n_{cp}^{η} = len(H) - (q+1) #No.of control points in η direction (6-(1+1) = 4)

3. Total number of control points for the geometry
   $n_{cp} = n_{cp}^{\xi} * n_{cp}^{\eta}$ = 3*4 = 12
   The control points are given by

i	$P_{i,0}$	$P_{i,1}$	$P_{i,2}$
0	$(0, 0, 0, 1)$	$(0, 5, 0, 1)$	$(0, 10, 0, 1)$
1	$(3.33, 0, 0, 1)$	$(3.33, 5, 0, 1)$	$(3.33, 10, 0, 1)$
2	$(6.67, 0, 0, 1)$	$(6.67, 5, 0, 1)$	$(6.67, 10, 0, 1)$
3	$(10, 0, 0, 1)$	$(10, 5, 0, 1)$	$(10, 10, 0, 1)$

   with fourth value in the paranthesis being weights of respective control points. A control point assembly array and knot vector connectivity matrix are used to connect the elements.

7.2.5 Results and discussions

Abaqus plane strain full integration piezoelectric element (**CPE4E**) is used for analysis. For the comparision between abaqus and IGA elements 2 elements, along x direction and 3 elements along y direction are used. A different number of elements are used along each direction in order to verify if the code generates proper results in unsymmetric conditions aswell w.r.t number of elements in each direction.
The below figures shows the values of displacements (U), electrical potentials (EPOT) and reaction forces (RF) for both abaqus and IGA elements.
Figure(25) and Figure(26) shows the displacement (U1) values of the CPE4E elements and IGA elements at 100 % loading in x direction respectively.

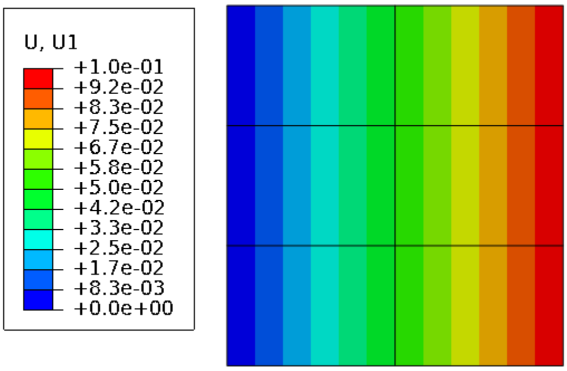


Figure 25: CPE4E Element:U1

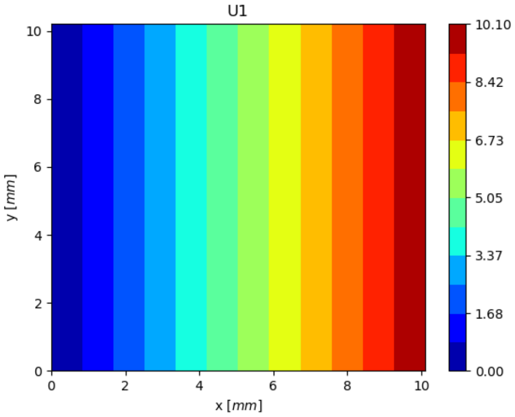


Figure 26: IGA Piezoelectric Element:U1

Figure(27) and Figure(28) shows the displacement (U2) values of the CPE4E elements and single IGA elements at 100 % loading in y direction respectively.

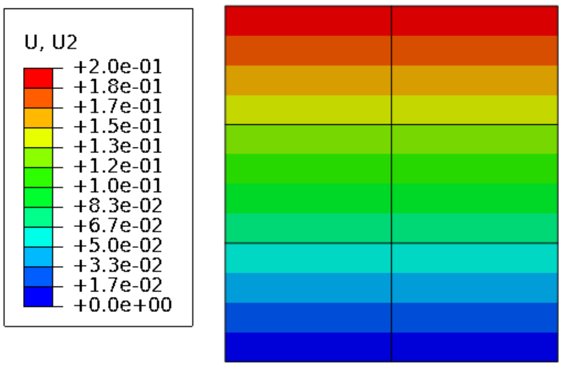


Figure 27: CPE4E Element:U2

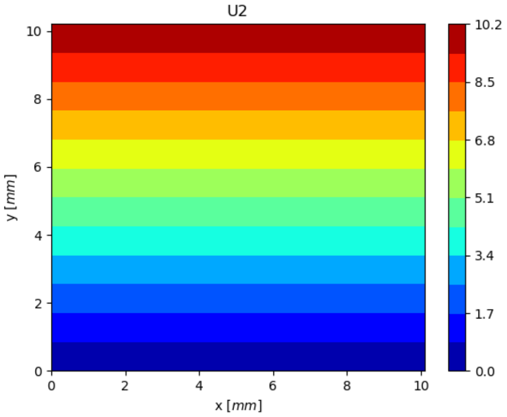


Figure 28: IGA Piezoelectric Element:U2

Figure(29) and Figure(30) shows the Reaction force values (RF1) of the CPE4E elements and IGA elements at 100 % loading in x direction respectively.

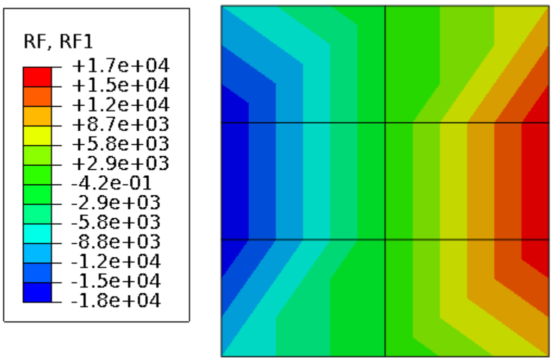


Figure 29: CPE4E Element:RF1

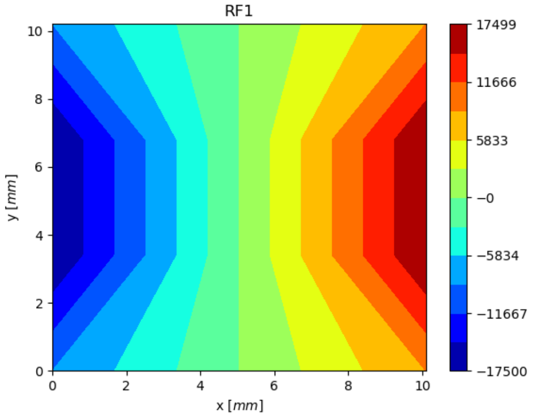


Figure 30: IGA Piezoelectric Element:RF1

Figure(31) and Figure(32) shows the Reaction force values (RF2) of the CPE4E elements and IGA elements at 100 % loading in y direction respectively.

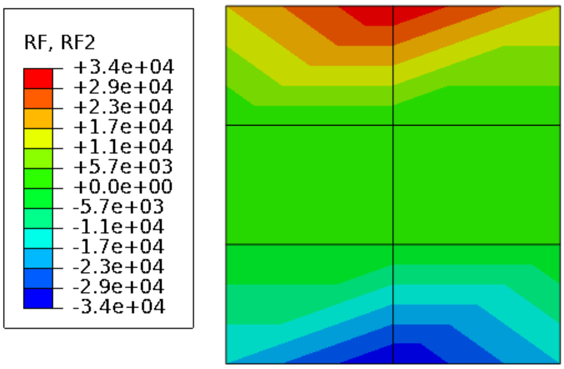


Figure 31: CPE4E Element:RF2

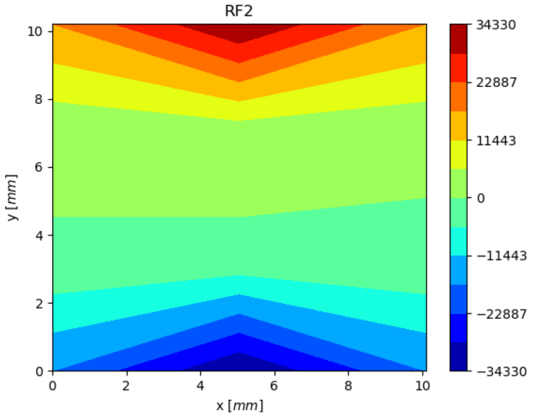


Figure 32: IGA Piezoelectric Element:RF2

Figure(33) and Figure(34) shows the Electrical potential values (EPOT) of the CPE4E elements and IGA elements at 100 % loading respectively.

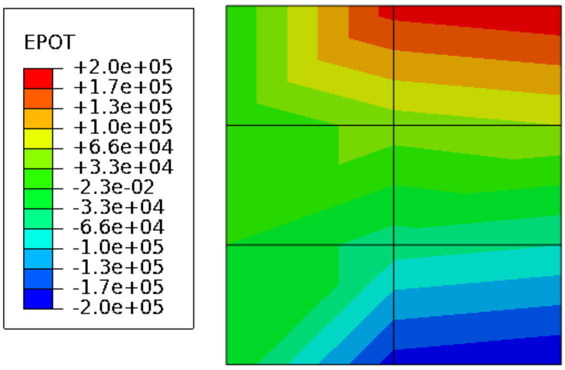


Figure 33: CPE4E Element:EPOT

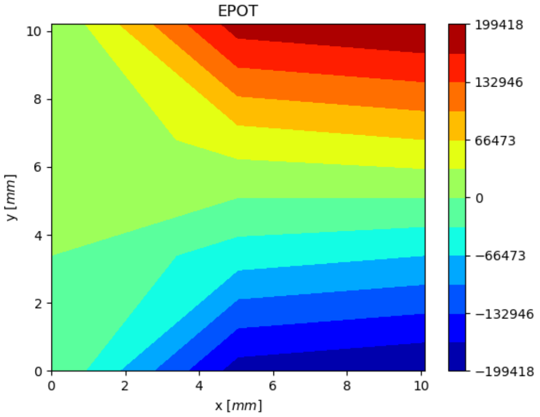


Figure 34: IGA Piezoelectric Element:EPOT

7.2.6 Conclusion

As shown in figures above the values generated by IGA code are inline with the results of Abaqus elements.

8 Milestones achieved

The following table describes the proposed coding activities and achieved activities.

Proposed Activities	Achieved
Implementation of Isogeometric Analysis	yes
IGA implementation for a Single element 2D case	yes
Coupling between mechanical and electrical Dofs	yes
Verification of results with abaqus inbuilt piezoelectric element	yes
Extra Activities	Achieved
Implementation of Knot and Control point assembly arrays	yes
Code extension to multiple elements	yes

References