Universität Stuttgart

Thermo-Elasticity for Large Deformations

Simulation Of Coupled Problems Using Finite Element Method

09.03.2020

Aniket Angre	3379505
Abigail Austin	3379550
Shlok Gupta	3379327
Ayush Pandey	3379369

Overview

- **Introduction**
- **Motivation- Use in industries**
- **Why Non Linear?**
- **Equation Formulation**
- **Problem Formulation**
- **Results:**
  - **Pure Mechanical Loading**
  - **Pure Thermal Loading**
  - **Mixed Loading**
  - **Mesh convergence study for Maximum displacement**

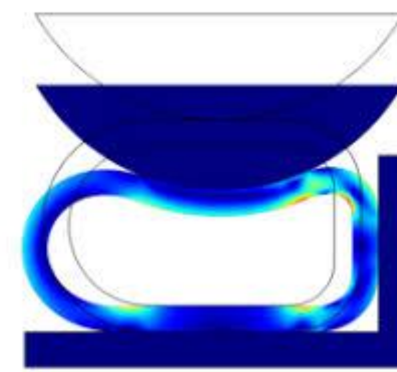
Introduction

Introduction

- Thermo-elasticity by definition addresses:
  - Mechanical deformation - Elastic deformation
  - Thermal effects - Temperature gradients, Heat flux
  - Their couplings - Thermal expansion

- Hyper-elastic materials posses certain characteristics:
  - Fully recoverable large elastic deformations in order of 100 to 700 %
  - They show a highly nonlinear stress-strain relation
    - ➢ Tension - The material softens and then stiffens again
    - ➢ Compression - A stiff response (incompressible)

Rubber



Hyper-elastic Seal

Motivation: Use in industries

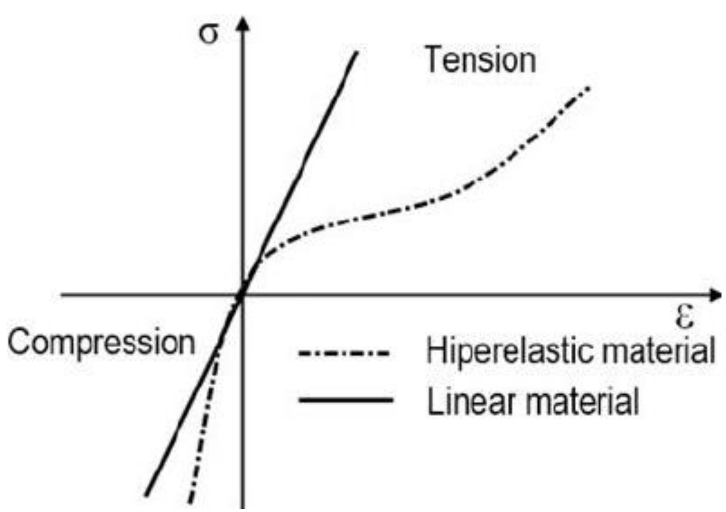
Motivation: Use in industries

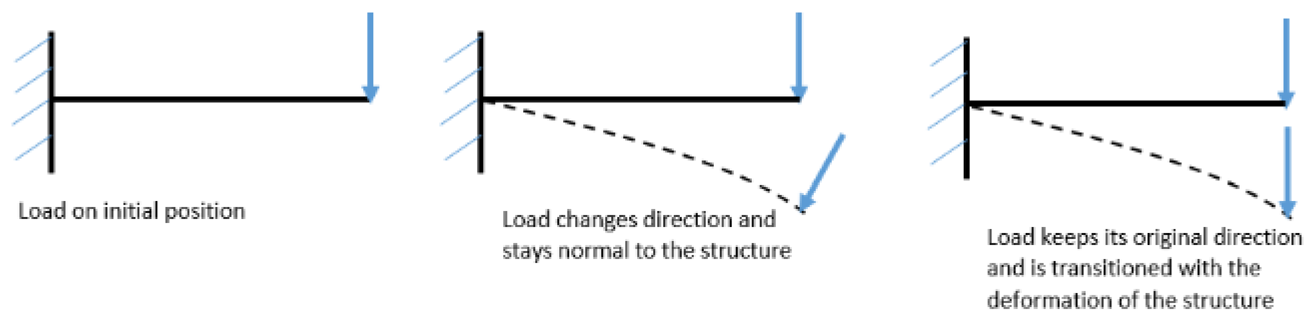
- Flexible adhesive joints, with very low elastic modulus and very large failure deformation
- Elastomeric pads in bridges, rail pads, car door seal, car tires
- Major stresses in Engine cylinder walls are due to temperature variation and mechanical impact
- Stresses experienced by a Pressure cooker are due to internal pressure and temperature variation
- Pressurized storage containers for liquified gases
- Industrial chimneys – Self weight, pressure and temperature variation
- Turbo machinery – High pressure leads to increase in temperature
- Casting and forging machinery
- Bird impact in aerospace

Why Non Linear?

Why Non Linear?

- Small vs large deformations
- Hyper-elastic materials (rubber and plastic) show a highly nonlinear stress-strain relation.
- Changes in geometry as the structure deforms are considered in formulating the constitutive and equilibrium equations.
- In case of contact we can have non linear boundary conditions (kinematic constraints).
    - ➢ As opposed to a linear static analysis, where the stiffness matrix remains constant, these effects result in a stiffness matrix which is not constant during the load application. As a result, a different solving strategy is required - Newton Raphson Solver.





Equation Formulation

Equation Formulation

General structure of balance relations

- **Local differential equations – valid at every material point**

- Balance of mass $\rho + \rho div\underline{\dot{x}} = 0$
- Balance of linear momentum $\int_{B_o}(Div\,\underline{P} + \rho_o \times (\underline{b} - \underline{\ddot{x}}))dV_o = \underline{0}$

  Assumption: Quasi-static : No acceleration → $\int_{B_o} Div\,\underline{P} dV_o = \underline{0}$
- Balance of moment of momentum $\underline{T} = \underline{T}^T$
- Balance of energy $\int_B \rho\dot{\epsilon}\, dv = \int_B \left(\underline{T}.\underline{D} + \rho r - div\underline{q}\right) dv$
- Entropy inequality (CDI) $-\rho\left(\dot{\varphi} + \dot{\theta}\eta\right) + \underline{T} \cdot \underline{D} - \frac{1}{\theta}\underline{q} \cdot grad\theta \geq 0$

Equation Formulation

Material Modelling

- **Thermodynamic consistent constitutive modelling**
- **Axioms of Thermodynamics**
  - Principle of determinisms
  - Principle of Equipresence
  - Principle of local action
  - Material Frame indifference
- **Evaluation of entropy inequality**
- **Helmholtz free energy function for isotropic hyper elastic materials**

$$\underline{T} = 2\rho \underline{F}\frac{\partial \varphi}{\partial \underline{C}}\underline{F}^T$$

$$\eta = -\frac{\partial \varphi}{\partial \theta}$$

$$\frac{\partial \varphi}{\partial grad\theta} = 0 \quad => \quad \varphi = \varphi(\underline{C}, \theta)$$

$$-\frac{1}{\theta}\underline{q} \cdot grad\theta \geq 0$$

$$\psi = \frac{1}{\rho_o}\left[\frac{1}{2}\lambda(\ln J)^2 - \mu lnJ + \frac{1}{2}\mu\left(\frac{I_c}{3}\right) - 3\,\alpha\kappa(lnJ)(\theta - \theta_o) - \rho_o c\left(\theta\ln\frac{\theta}{\theta_o} - \theta + \theta_o\right)\right]$$

Where, $\lambda, \mu \rightarrow$ Material constants (Lame constants)

$\alpha \rightarrow$ Thermal expansion coeffient , c $\rightarrow$ Heat capacity

$\kappa \rightarrow$ Compression Modulus , $I_c \rightarrow tr\,\underline{C}$

Equation Formulation

Balance of Momentum

- **Weak Form:**

$$\int_{B_o} Div\, \underline{P}.\, \delta \underline{u} dV_o = \underline{0}$$

Final Form → $\int_{B_o} \underline{S}.\, \delta \underline{E} dV_o - \int_{\partial B_o} \underline{t_0}.\, \delta \underline{u} dA_o = 0$

- **Discretized Weak Form**

$$G_u = \sum^{J} \delta \underline{du}^I \left[\int_{B_o} B^T(\underline{u}, \underline{X}) \cdot \underline{S} dV_o - \int_{\partial B_o} N^I \underline{t} dA_o \right]$$

- **Linearized Weak Form**

$$\Delta D G_u = \sum^I \sum^J \delta \underline{du}^I \left[\int_{B_0} \left(B^{T^I}(\underline{u}, \underline{X}) . \frac{\partial \underline{S}}{\partial \underline{E}} . B^J(\underline{u}, \underline{X}) \right) dV_o \Delta\, d\underline{u}^J \right] - \sum^I \sum^J \delta \underline{du}^I \left[\int_{B_0} \left(B^{T^I}(\underline{u}, \underline{X}) . \frac{\partial \underline{S}}{\partial \theta} . N^J(\underline{X}) \right) dV_o \Delta\, d\theta^J \right]$$

Equation Formulation

Balance of Momentum

- $\underline{S} = \mu\left(\underline{I} - \underline{C}^{-1}\right) + \lambda \ln(J)\, \underline{C}^{-1} - 3\alpha_s k(\theta - \theta_0)\underline{C}^{-1}$

- $\frac{d\underline{S}}{d\underline{E}} = -\mu \frac{d\underline{C}^{-1}}{d\underline{E}} - \lambda lnJ \frac{d\underline{C}^{-1}}{d\underline{E}} - \lambda \frac{dlnJ}{d\underline{E}} \otimes \underline{C}^{-1} - 3\alpha_s \kappa(\theta - \theta_o) \frac{d\underline{C}^{-1}}{d\underline{E}}$

  The derivatives → $\frac{d\underline{C}^{-1}}{d\underline{E}} = -\left[C_{lm}^{-1} C_{nk}^{-1} + C_{ln}^{-1} C_{mk}^{-1}\right]$

  $\frac{dlnJ}{d\underline{E}} \otimes \underline{C}^{-1} = \underline{C}^{-1} \otimes \underline{C}^{-1}$

  $$\frac{dS_{lk}}{dE_{mn}} = [\mu - \lambda lnJ + 3\alpha_s \kappa(\theta - \theta_o)]\left[C_{lm}^{-1} C_{nk}^{-1} + C_{ln}^{-1} C_{mk}^{-1}\right] + \lambda C_{lk}^{-1} C_{mn}^{-1}$$

- $\frac{\partial S}{\partial \theta} = -3\alpha_s k \underline{C}^{-1}$

Equation Formulation

Balance of Energy

- **Weak Form:**

$$\int_{B_o} \underline{q_o} \,.\, Grad\delta\theta dV_o - \int_{B_o} \left(\rho_o c\dot{\theta} + 3\alpha_s\kappa\theta\underline{D}.\underline{I} - \rho_o r\right)\delta\theta dV_o = \int_{\partial B_o} \underline{q_0} \,.\, \underline{N}\delta\theta dA_o$$

Where ,

$$q_o = -J\alpha_\theta Grad\theta\underline{C}^{-1}$$

- **Discretized Weak Form**

$$G_\theta = \sum^I \delta d\theta^I \left[\int_{B_o} B^I\left(\underline{u,X}\right)\cdot q_{0,lin} dV_o - \int_{B_o} N^I\rho_0 C\theta dV_o - \int_{B_o} N^I 3\alpha_s\kappa\theta tr\underline{\dot{E}} dV_o - \int_{B_o} N^I\rho_0 r dV_o - \int_{B_o} N^I q_{0,lin}\underline{N} dA_o\right]$$

- **Linearized Weak Form**

$$\Delta DG_\theta = \sum^I\sum^J \delta d\theta^I \int_{B_o}\left(B^{T^I}(\theta).\frac{\partial q_{\{o,lin\}}}{\partial Grad\theta}.B^J(\theta)\right)dV_o\Delta\, d\theta^J - \sum^I\sum^J \delta d\theta^I \int_{B_o}\left(N^I\frac{\partial Bal\ Energy}{\partial\dot{\theta}}\frac{\delta}{\beta\Delta t}.N^J\right)dV_o\Delta\, d\theta^J$$
$$- \sum^I\sum^J \delta d\theta^I \int_{B_o}\left(N^I\frac{\partial Bal\ Energy}{\partial\theta}N^J\right)dV_o\Delta\, d\theta^J - \sum^I\sum^J \delta\theta^I \int_{B_o}\left(N^I\frac{\partial Bal\ Energy}{\partial\underline{\dot{E}}}\frac{\delta}{\beta\Delta t}.B^J(\underline{u},\underline{X})\right)dV_o\Delta\, d\underline{u}^J$$

Equation Formulation

Balance of Energy

- $\left(\frac{\partial q_{0,lin}}{\partial Grad\theta}\right) = -\alpha_\theta \underline{I}$
- $\frac{\partial Bal\ Energy}{\partial \dot{\theta}} = \rho_0 C$
- $\frac{\partial Bal\ Energy}{\partial \theta} = 3\alpha_s k \cdot tr\dot{E}$
- $\frac{\partial Bal\ Energy}{\partial \underline{\dot{E}}} = 3\alpha_s k\theta \underline{I}$

- **Time Discretization: Newmark Beta**

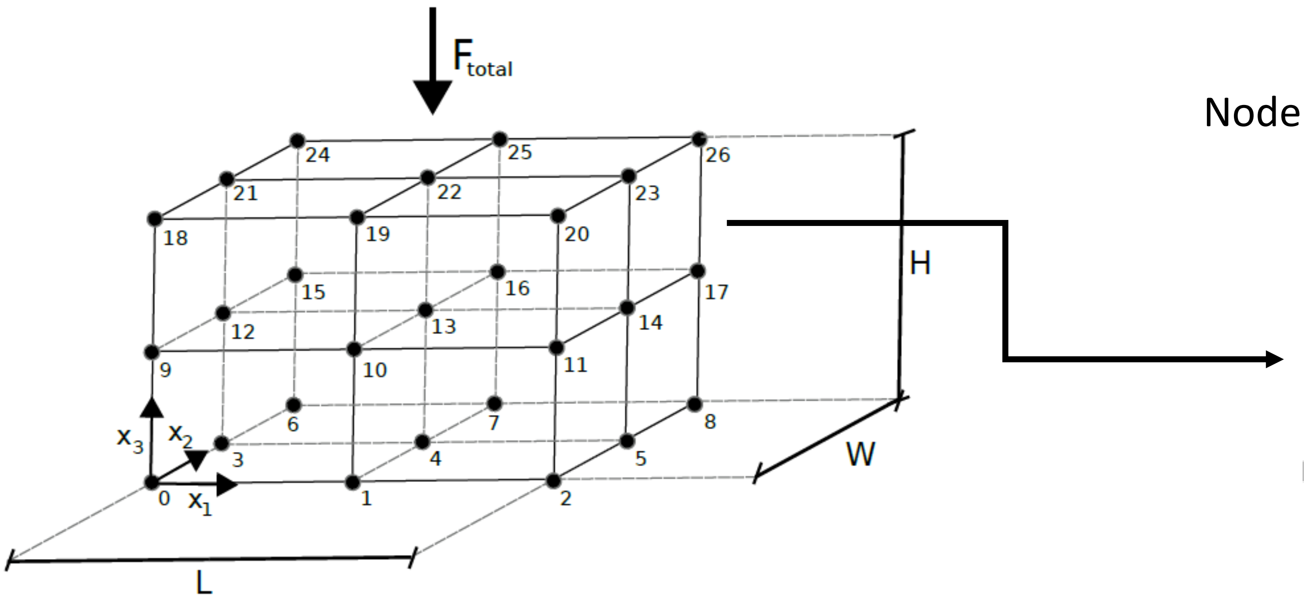
$$\underline{\dot{X}}_{t+\Delta t} = \underline{\dot{X}} + \left[(1-\delta)\underline{\ddot{X}}_t + \delta\left(\underline{\ddot{X}}_{t+\Delta t}\right)\right]\Delta t \qquad \Delta\underline{\dot{x}} = \frac{\delta}{\beta\Delta t}\Delta\underline{x}$$

$$\underline{X}_{t+\Delta t} = \underline{X} + \underline{\dot{X}}_t\Delta t + \left[\left(\frac{1}{2}-\beta\right)\underline{\ddot{X}}_t + \beta\left(\underline{\ddot{X}}_{t+\Delta t}\right)\right]\Delta t^2 \qquad \Delta\underline{\ddot{x}} = \frac{1}{\beta\Delta t^2}\Delta\underline{x}$$

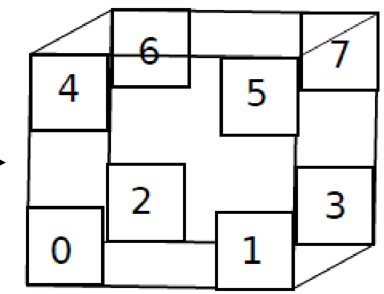
Problem Formulation

Problem Formulation

Finite Element Mesh, Boundary Conditions and Loads



Node Numbering Sequence




```
fixedDofs = {\
        0 : [0, 1, 2], \
        1 : [1, 2], \
        2 : [1, 2], \
        3 : [0, 2], \
        4 : [2], \
        5 : [2], \
        6 : [0, 2], \
        7 : [2], \
        8 : [2]}
```

```
externalForces = {\
        18 : [0, 0, fz/4/4, heatFlux / 4 / 4],
        19 : [0, 0, fz/4/2, heatFlux / 4 / 2],
        20 : [0, 0, fz/4/4, heatFlux / 4 / 4],
        21 : [0, 0, fz/4/2, heatFlux / 4 / 2],
        22 : [0, 0, fz/4/1, heatFlux / 4 / 1],
        23 : [0, 0, fz/4/2, heatFlux / 4 / 2],
        24 : [0, 0, fz/4/4, heatFlux / 4 / 4],
        25 : [0, 0, fz/4/2, heatFlux / 4 / 2],
        26 : [0, 0, fz/4/4, heatFlux / 4 / 4]}
```

Problem Formulation

Equation System:

$$\Delta DG_u = \sum^{I}\sum^{J} \delta \underline{u}^I \left[\int_{B_0} \left(B^{T^I}(\underline{u},\underline{X}).\frac{\partial \underline{S}}{\partial \underline{E}}.B^J(\underline{u},\underline{X})\right) dV_o \Delta\, d\underline{u}^J\right] - \sum^{I}\sum^{J} \delta \underline{u}^I \left[\int_{B_0} \left(B^{T^I}(\underline{u},\underline{X}).\frac{\partial \underline{S}}{\partial \theta}.N^J(\underline{X})\right) dV_o \Delta\, d\theta^J\right]$$

$$[\boldsymbol{\delta d\underline{u}} \quad \boldsymbol{\delta d\theta}]\left\{\begin{bmatrix} \boldsymbol{\underline{K}_{uu}} & \boldsymbol{\underline{K}_{u\theta}} \\ \boldsymbol{\underline{K}_{\theta u}} & \boldsymbol{K_{\theta\theta}} \end{bmatrix}\begin{bmatrix} \boldsymbol{\Delta d\underline{u}} \\ \boldsymbol{\Delta d\theta} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\underline{R}_u} \\ \boldsymbol{R_\theta} \end{bmatrix}\right\} = \underline{\boldsymbol{0}}$$

$$\Delta DG_\theta = \sum^{I}\sum^{J} \delta d\theta^I \int_{B_o} \left(B^{T^I}(\theta).\frac{\partial q_{\{o,lin\}}}{\partial Grad\theta}.B^J(\theta)\right) dV_o \Delta\, d\theta^J$$

$$- \sum^{I}\sum^{J} \delta d\theta^I \int_{B_o} \left(N^I \rho_0 C\, \frac{\delta}{\beta \Delta\, t}.N^J\right) dV_o \Delta\, d\theta^J$$

$$- \sum^{I}\sum^{J} \delta d\theta^I \int_{B_o} (N^I 3\alpha_s \kappa tr \dot{E} N^J) dV_o \Delta\, d\theta^J$$

$$- \sum^{I}\sum^{J} \delta\theta^I \int_{B_o} \left(N^I \frac{\delta}{\beta \Delta t} 3\alpha_s \kappa \theta \underline{I}.B^J(\underline{u},\underline{X})\right) dV_o \Delta\, d\underline{u}^J$$

Problem Formulation

Implementation of Finite Element Code

- Shape functions: Tri-Linear Functions

```
def shapeFunction(self, position, point):
    return 1.0/8.0 * (1.0 + position[0]*point[0]) * (1.0 + position[1]*point[1]) * (1.0 + position[2]*point[2])
```

- Green Lagrange strain tensor

$$\boxed{E_{ij} = \frac{1}{2}\left(u_{i,j} + u_{j,i} + u_{k,i}\, u_{k,j}\right)}$$

$$\boldsymbol{E}(\boldsymbol{u}) = \begin{bmatrix} E_{11} \\ E_{22} \\ E_{33} \\ 2E_{12} \\ 2E_{23} \\ 2E_{13} \end{bmatrix} = \begin{bmatrix} u_{1,1} \\ u_{2,2} \\ u_{3,3} \\ u_{1,2} + u_{2,1} \\ u_{2,3} + u_{3,2} \\ u_{1,3} + u_{3,1} \end{bmatrix} + \begin{bmatrix} 1/2\ (u_{1,1}\, u_{1,1} + u_{2,1}\, u_{2,1} + u_{3,1}\, u_{3,1}) \\ 1/2\ (u_{1,2}\, u_{1,2} + u_{2,2}\, u_{2,2} + u_{3,2}\, u_{3,2}) \\ 1/2\ (u_{1,3}\, u_{1,3} + u_{2,3}\, u_{2,3} + u_{3,3}\, u_{3,3}) \\ u_{1,1}\, u_{1,2} + u_{2,1}\, u_{2,2} + u_{3,1}\, u_{3,2} \\ u_{1,2}\, u_{1,3} + u_{2,2}\, u_{2,3} + u_{3,2}\, u_{3,3} \\ u_{1,1}\, u_{1,3} + u_{2,1}\, u_{2,3} + u_{3,1}\, u_{3,3} \end{bmatrix}$$

- Differential operator (L) which maps the displacement vector variation to the Green Lagrange strain by addition of constant part and deformation-dependent part

- B = LN →

  Here L = D

$$\mathbf{D}_{\varepsilon} = \begin{bmatrix} \frac{\partial}{\partial X_1} & 0 & 0 \\ 0 & \frac{\partial}{\partial X_2} & 0 \\ 0 & 0 & \frac{\partial}{\partial X_3} \\ \frac{\partial}{\partial X_2} & \frac{\partial}{\partial X_1} & 0 \\ 0 & \frac{\partial}{\partial X_3} & \frac{\partial}{\partial X_2} \\ \frac{\partial}{\partial X_3} & 0 & \frac{\partial}{\partial X_1} \end{bmatrix} + \begin{bmatrix} u_{1,1}\frac{\partial}{\partial X_1} & u_{2,1}\frac{\partial}{\partial X_1} & u_{3,1}\frac{\partial}{\partial X_1} \\ u_{1,2}\frac{\partial}{\partial X_2} & u_{2,2}\frac{\partial}{\partial X_2} & u_{3,2}\frac{\partial}{\partial X_2} \\ u_{1,3}\frac{\partial}{\partial X_3} & u_{2,3}\frac{\partial}{\partial X_3} & u_{3,3}\frac{\partial}{\partial X_3} \\ u_{1,2}\frac{\partial}{\partial X_1} + u_{1,1}\frac{\partial}{\partial X_2} & u_{2,2}\frac{\partial}{\partial X_1} + u_{2,1}\frac{\partial}{\partial X_2} & u_{3,2}\frac{\partial}{\partial X_1} + u_{3,1}\frac{\partial}{\partial X_2} \\ u_{1,3}\frac{\partial}{\partial X_2} + u_{1,2}\frac{\partial}{\partial X_3} & u_{2,3}\frac{\partial}{\partial X_2} + u_{2,2}\frac{\partial}{\partial X_3} & u_{3,3}\frac{\partial}{\partial X_2} + u_{3,2}\frac{\partial}{\partial X_3} \\ u_{1,3}\frac{\partial}{\partial X_1} + u_{1,1}\frac{\partial}{\partial X_3} & u_{2,3}\frac{\partial}{\partial X_1} + u_{2,1}\frac{\partial}{\partial X_3} & u_{3,3}\frac{\partial}{\partial X_1} + u_{3,1}\frac{\partial}{\partial X_3} \end{bmatrix}$$

Problem Formulation

Implementation of Finite Element Code – Important snippets

```
# Implementation of non linear S
self.sigma[gp][0] = self.lmu * (1 - invCauchy[0][0]) + self.llambda * log(J) * invCauchy[0][0] \
                  - 3.0 * self.alphaS * self.kappa * (self.theta[gp] - self.theta0[gp]) * invCauchy[0][0]
self.sigma[gp][1] = self.lmu * (1 - invCauchy[1][1]) + self.llambda * log(J) * invCauchy[1][1] \
                  - 3.0 * self.alphaS * self.kappa * (self.theta[gp] - self.theta0[gp]) * invCauchy[1][1]
self.sigma[gp][2] = self.lmu * (1 - invCauchy[2][2]) + self.llambda * log(J) * invCauchy[2][2] \
                  - 3.0 * self.alphaS * self.kappa * (self.theta[gp] - self.theta0[gp]) * invCauchy[2][2]
self.sigma[gp][3] = - self.lmu * 2.0 * invCauchy[0][1] + self.llambda * log(J) * 2.0 * invCauchy[0][1] \
                  - 3.0 * self.alphaS * self.kappa * (self.theta[gp] - self.theta0[gp]) * 2.0 * invCauchy[0][1]
self.sigma[gp][4] = - self.lmu * 2.0 * invCauchy[1][2] + self.llambda * log(J) * 2.0 * invCauchy[1][2] \
                  - 3.0 * self.alphaS * self.kappa * (self.theta[gp] - self.theta0[gp]) * 2.0 * invCauchy[1][2]
self.sigma[gp][5] = - self.lmu * 2.0 * invCauchy[0][2] + self.llambda * log(J) * 2.0 * invCauchy[0][2] \
                  - 3.0 * self.alphaS * self.kappa * (self.theta[gp] - self.theta0[gp]) * 2.0 * invCauchy[0][2]
```

```
# Loop for implementation of linear entries
self.epsilon[gp][0] = self.gradU[gp][0][0]
self.epsilon[gp][1] = self.gradU[gp][1][1]
self.epsilon[gp][2] = self.gradU[gp][2][2]
self.epsilon[gp][3] = self.gradU[gp][0][1] + self.gradU[gp][1][0]
self.epsilon[gp][4] = self.gradU[gp][1][2] + self.gradU[gp][2][1]
self.epsilon[gp][5] = self.gradU[gp][2][0] + self.gradU[gp][0][2]
# Loop to add all non linear entrieds into E
for j in range(0,3):
    self.epsilon[gp][0] += 0.5 * self.gradU[gp][j][0] * self.gradU[gp][j][0]
    self.epsilon[gp][1] += 0.5 * self.gradU[gp][j][1] * self.gradU[gp][j][1]
    self.epsilon[gp][2] += 0.5 * self.gradU[gp][j][2] * self.gradU[gp][j][2]
    self.epsilon[gp][3] += self.gradU[gp][j][0] * self.gradU[gp][j][1]
    self.epsilon[gp][4] += self.gradU[gp][j][1] * self.gradU[gp][j][2]
    self.epsilon[gp][5] += self.gradU[gp][j][0] * self.gradU[gp][j][2]
```

```
# Linear theory - Heat flux is deformation independent
self.qlin[gp][0] = -self.alphaT * self.gradTheta[gp][0]
self.qlin[gp][1] = -self.alphaT * self.gradTheta[gp][1]
self.qlin[gp][2] = -self.alphaT * self.gradTheta[gp][2]
```

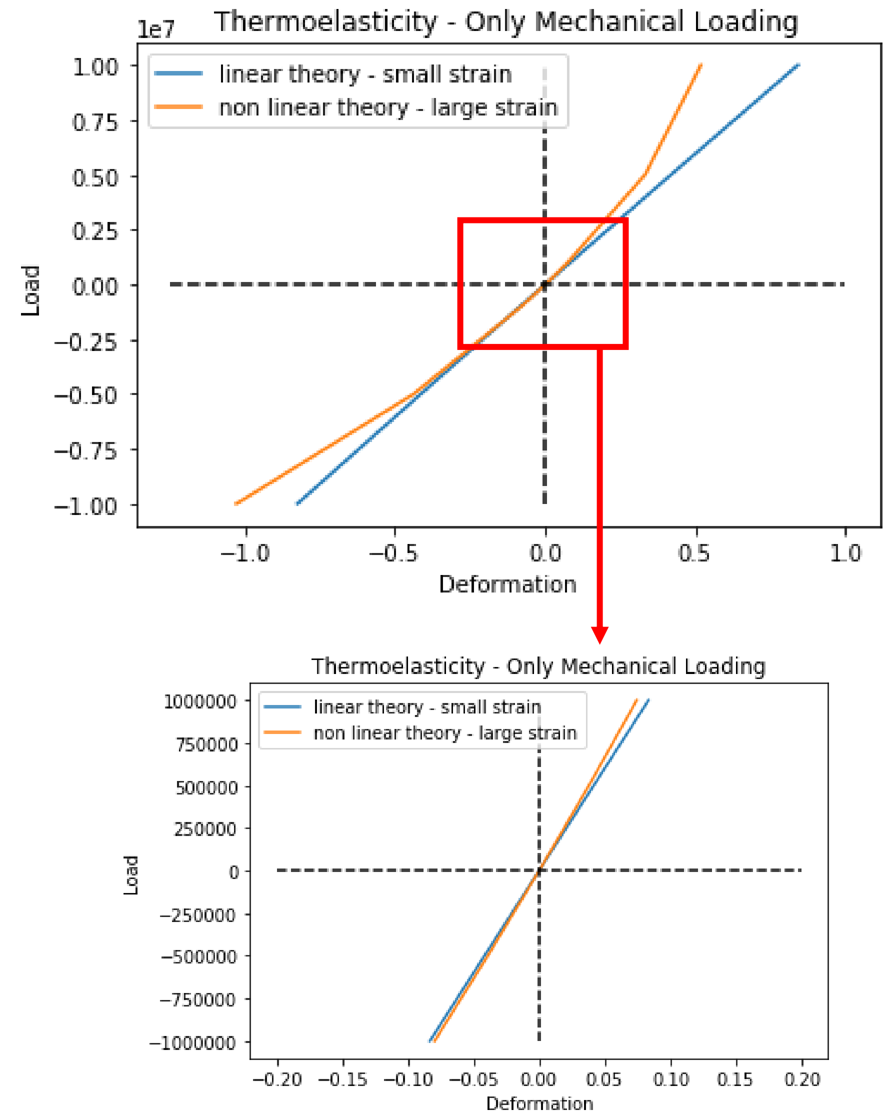
```
#  mC = ( lmu - llambda * ln(J) + 3.0 * alphaS * kappa * (theta - theta0)) * (C^-1 x C^-1)^T + llambda * (C^-1 x C^-1)
invCinvCT = zeros((3, 3, 3, 3), dtype=float64)
invCinvC = zeros((3, 3, 3, 3), dtype=float64)
mCconst1 = self.lmu - self.llambda * log(J) + 3.0 * self.alphaS * self.kappa * (self.theta[gp] - self.theta0[gp])
mCconst2 = self.llambda

# Formulation of (C^-1 x C^-1)^T and (C^-1 x C^-1)
index = [[0,0], [1,1], [2,2], [0,1], [1,2], [0,2]]

for l in range (0,3):               # l = p
    for k in range (0,3):           # k = q
        for m in range (0,3):       # m = r
            for n in range (0,3):   # n = s
                invCinvCT[l][k][m][n] = invCauchy[l][m] * invCauchy[n][k] + invCauchy[l][n] * invCauchy[m][k]
                invCinvC[l][k][m][n] = invCauchy[l][k] * invCauchy[m][n]

# Formulation of mC which is changing at every gauss point of the element
for i in range (0,6):
    for j in range (0,6):
        self.mC[gp][i][j] = mCconst1 * invCinvCT[index[i][0]][index[i][1]][index[j][0]][index[j][1]] \
                          + mCconst2 * invCinvC[index[i][0]][index[i][1]][index[j][0]][index[j][1]]
```

Results

Results

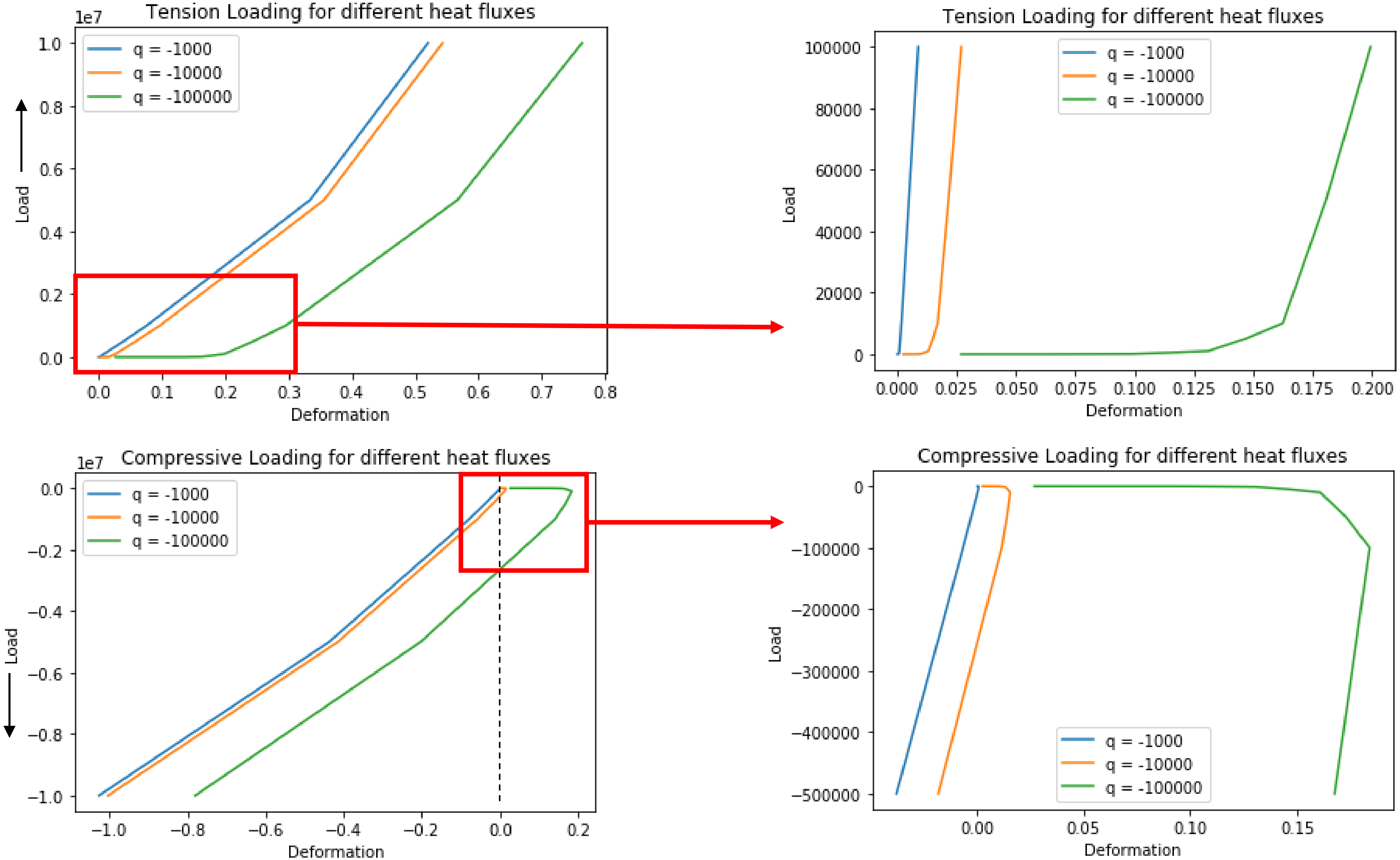
Load vs. Deformation – Pure Mechanical Loading

- Small strain theory → linearised E and q
  → linear results throughout
- Large strain theory → non linear E formulation
  → additional geometric stiffness
- Tensile Load → deformation is reducing in comparison with the linear theory
- Compressive Load → non linearity of curve is not pronounced



Results

Load vs. Deformation – Mixed Loading



Results

Load vs. Deformation – Mixed Loading

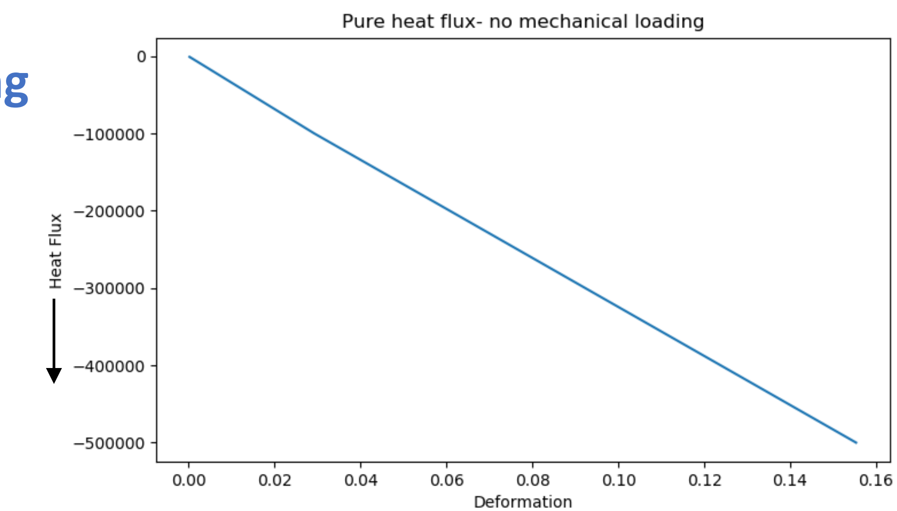
- Thermo-elasticity with small strains considers linearised E and linearised q.

α	LINEAR		NON-LINEAR	
	Z - displacement			
	1st Load Step	Last Load Step	1st Load Step	Last Load Step
0.0001	-0.00517115	0.124248263	-0.0052514249	0.1256020314
0.00001	-0.007902588	0.0179542813	-0.0077980955	0.0178963899
0	-0.0083325397	-0.0083325390	-0.0083321567	-0.0083321568

- Fz = -100000 & q = -500000

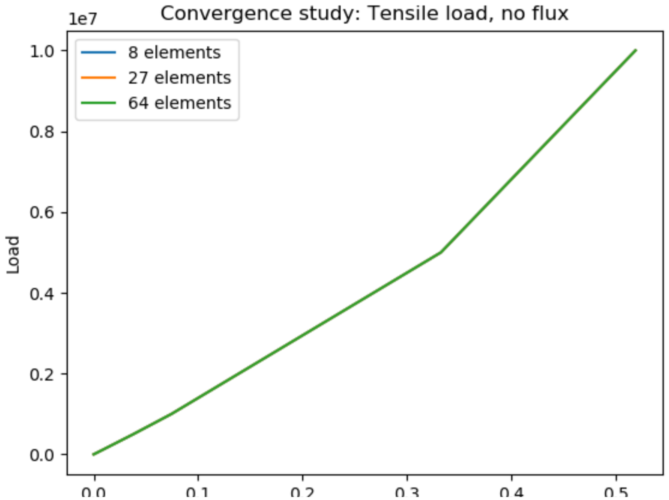
Heat Flux vs. Deformation – Pure Thermal Loading

- We get a linear graph → linear heat flux theory.

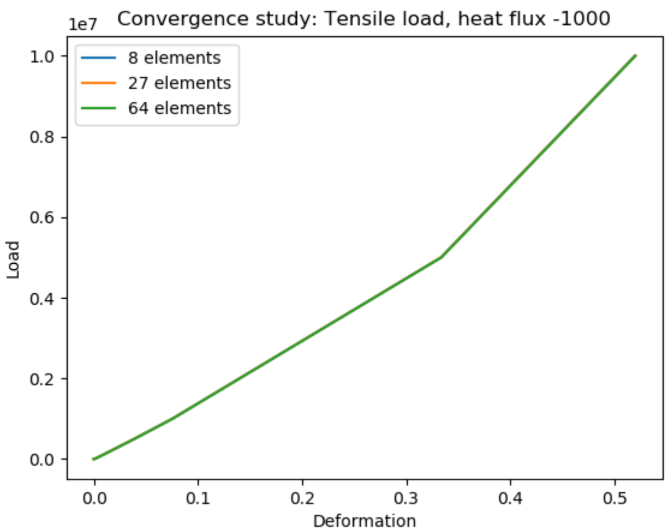


Results

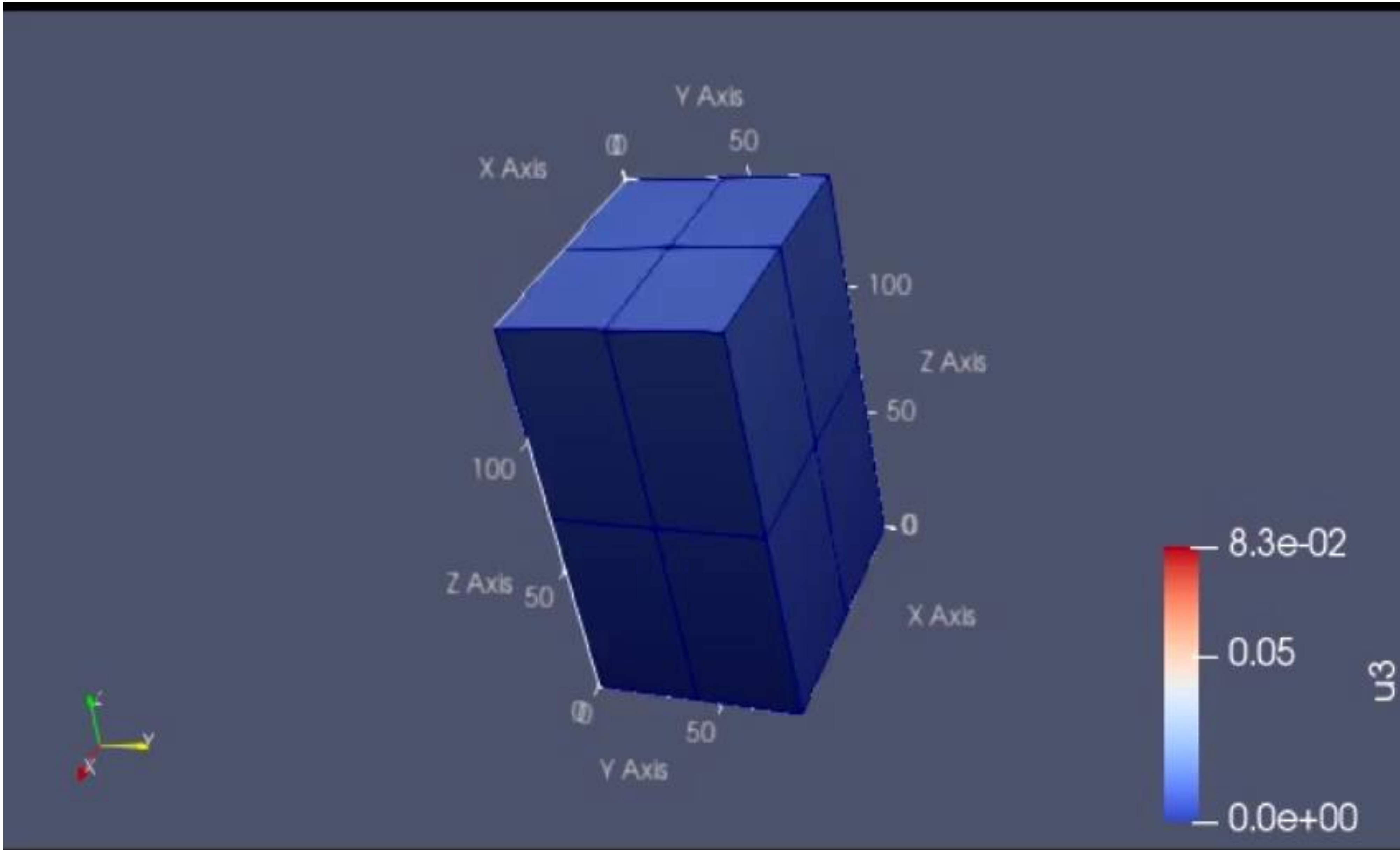
Mesh Convergence Study



Load	8 Elements	27 Elements	64 Elements
1e6	0.0744642054	0.0744597857	0.0744597857
5e6	0.3323761857	0.332364331	0.332364331
1e7	0.5184983087	0.5185321238	0.5185690033



Load	8 Elements	27 Elements	64 Elements
1e6	0.0754666549	0.0756117875	0.0760184981
5e6	0.3335196949	0.3336648635	0.3340613036
1e7	0.5197699119	0.5199717548	0.5203819076


Y Axis
50
X Axis
100
Z Axis
50
0
X Axis
100
Z Axis
50
50
Y Axis
8.3e-02
0.05
0.0e+00
u3

Universität Stuttgart

Vielen Dank!

Aniket Angre	3379505
Abigail Austin	3379550
Shlok Gupta	3379327
Ayush Pandey	3379369