Technische Universität Bergakademie Freiberg

PERSONAL PROGRAMMING PROJECT
(PPP)

Implementation of Gradient Elasticity Model In FEM

Dhaval Rasheshkumar Patel

63940

Supervised by
Dr. Sergii Kozinov

February 12, 2020

Abstract

List of Figures

Contents

1 Introduction

Classical continuum solid mechanics theories, such as linear or non-linear elasticity and plasticity, have been used in wide range of fundamental problems and applications in various fields, but "Classical Continuum Constitutive Models" possess no material/intrinsic scale. So in this regime of micron and nano-scales that experimental evidence and observations have suggested that classical continuum theories do not suffice for an accurate and detailed description in the modelling of size dependent phenomena. Moreover, classical elastic singularities as those emerging during the application of point loads and description of size effects also not solve by it.

In General, in Gradient elasticity theories, the length scales enter the constitutive equations through the elastic strain energy function, which, in this case depends not only on the strain tensor but also on gradients of the strain tensors Gradient Elasticity theories provide extensions of the classical equations of elasticity with additional higher order spatial derivatives of strains and stresses, but in Gradient elasticity theory one of the most challenging task is to keep the number of additional constitutive parameters to a minimum.

Stress equation of equilibrium, constitutive equations and boundary conditions of the "Strain Gradient theory" were first given in a non-linear form by Toupin in 1960s. After this famous Strain Gradient elasticity theory in three different forms proposed by Mindlin in 1960s. In such theories, when the problem is formulated in terms of displacements, the governing partial differential equation contains the fourth order derivative of displacements. If traditional finite element formulation are used for the numerical solution of such problems, then C^1 displacement continuity is required. C^1 displacement continuity means displacement and its first derivative is continuous in inter-element. However, in FEM there is no any robust C^1 continuous element. So an alternative mixed finite elements formulation is developed, in which both the displacement and the displacement gradients are used as independent unknowns and their relationship is enforced using Lagrange Multiplier Method. In 1999 JOHN Y. SHU proposed the mixed finite formulation based ToupinMindlin theories, in which only C^0 continuous element was used. After this in 2002 E. Amanatidou and N. Aravas also proposed the Mixed type FEM formulation for ToupinMindlin theories and also used C^0 continuous element.

In the present "Personal Programming Project" report, in the very first section we discussed the theoretical background of the strain gradient elasticity theory with its evolutions and different versions. Following to this section, we discussed in brief few of them theories. After this the Mixed type of FEM formulation with C^0 continuous element for Strain Gradient theory is discussed in details.

2   Theoretical Background

In this Section the main focus is to understand the "Classical Strain Gradient Elasticity Theory" and motivation behind the evolution of Strain Gradient Elasticity theories. In this section the brief overview of Toupin's, Mindlin's, Eringen's and Aifantis theory is given. After this also discussed about different possible FEM formulation.
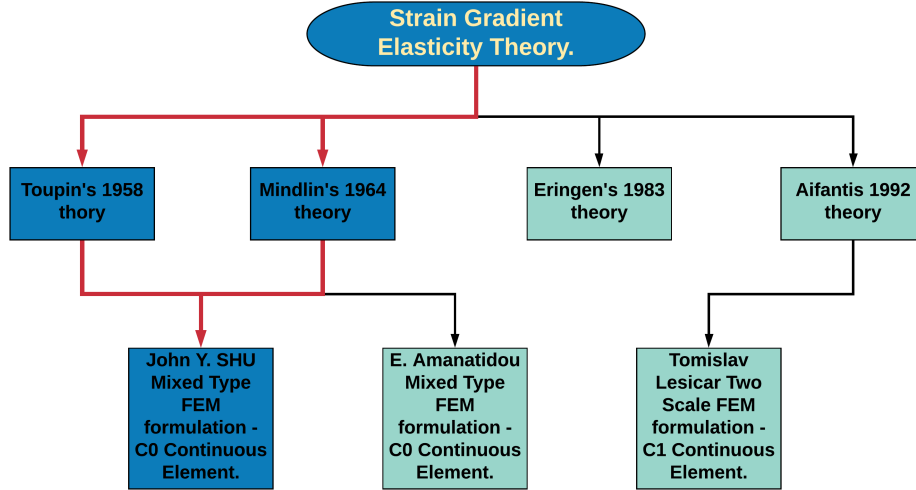
Figure 1: The history of Strain Gradient Theory

2.1   Mindlin's 1964 Theory.

In the early 1960s, the Mindlin presented [4] a theory of elasticity with Microstructure length, in which he distinguished between kinematic quantities on two scale micro and macro. As discussed in introduction that it is very challenging task to keep the number of constitutive parameters to a minimum. In this theory constitutive tensor contains 1764 coefficients in total, but for isotropic material its reduced drastically to amount 18. Mindlin present the Strain energy equation as,

$$
\begin{aligned}
U = &\frac{1}{2}\lambda\varepsilon_{ii}\varepsilon_{jj} + \mu\varepsilon_{ij}\varepsilon_{ij} + \frac{1}{2}b_1\gamma_{ii}\gamma_{jj} + \frac{1}{2}b_2\gamma_{ij}\gamma_{ij} + \frac{1}{2}b_3\gamma_{ij}\gamma_{ji} + g_1\gamma_{ii}\varepsilon_{jj} \\
&+ g_2(\gamma_{ij} + \gamma_{ji})\varepsilon_{jj} + a_1\kappa_{iik}\kappa_{kjj} + a_2\kappa_{iik}\kappa_{jkj} + \frac{1}{2}a_3\kappa_{iik}\kappa_{jjk} \\
&+ \frac{1}{2}a_4\kappa_{iij}\kappa_{ikk} + a_5\kappa_{iij}\kappa_{kik} + \frac{1}{2}a_8\kappa_{iji}\kappa_{kjk} + \frac{1}{2}a_{10}\kappa_{ijk}\kappa_{ijk} \\
&+ a_{11}\kappa_{ijk}\kappa_{jki} + \frac{1}{2}a_{13}\kappa_{ijk}\kappa_{ikj} + \frac{1}{2}a_{14}\kappa_{ijk}\kappa_{jik} + \frac{1}{2}a_{15}\kappa_{ijk}\kappa_{kji}
\end{aligned}
\tag{1}
$$

where λ and μ are the usual Lame constants and the various a_i , b_i and g_i are 16 additional constitutive coefficients. However for practical purpose the use of eq(1) is very limited as it requires so much additional coefficients. In later 1960s Mindlin also formulated the simpler version of his own elasticity theory by making assumption of expressing the strain energy density in terms of displacement only. So in last he proposed three different form of it. However, the equation of strain energy density is of order four and it requires C^1 continuity.

2.2 Aifanti's 1992 Theory.

In the early 1990s, motivated by own work in plasticity and non-linear elasticity, Aifantis [2] suggested to extend the linear constitutive model given by Mindlin in form II, in which second order terms are expressed as the strain gradient tensor. So he modified this Mindlin form II by neglecting the most of the gradient coefficients. Mindlin present that for a general isotropic elastic solid the strain energy density depends upon ε_{ij} and κ_{ijk} as the following,

$$
\begin{aligned}
W(\varepsilon, \kappa) =& G(\varepsilon_{ij}\varepsilon_{ij} + \frac{\nu}{1-2\nu}\varepsilon_{ij}\varepsilon_{ij}) \\
&+ a_1\kappa_{iik}\kappa_{kjj} + a_2\kappa_{iik}\kappa_{kjj} + a_3\kappa_{iik}\kappa_{kjj} + a_4\kappa_{iik}\kappa_{kjj} + a_5\kappa_{iik}\kappa_{kjj}
\end{aligned}
\tag{2}
$$

where G is elastic shear modulus, ν is Poisson's ratio and a_1, a_2, a_3, a_4, a_5 are material constants.

In this he considered,

$$
a_1 = a_3 = a_5 = 0, \quad a_2 = \frac{\nu}{1-2\nu}Gl^2, \quad a_4 = Gl^2
\tag{3}
$$

Accordingly, the strain energy density is defined as

$$
W = \frac{1}{2}\lambda\varepsilon_{ii}\varepsilon_{jj} + \mu\varepsilon_{ij}\varepsilon_{ij} + l^2(\frac{1}{2}\lambda\varepsilon_{ii,k}\varepsilon_{jj,k} + \mu\varepsilon_{ij,k}\varepsilon_{ij,k})
\tag{4}
$$

In Eq. (4), λ and μ are Lame constants, while l represents microstructural parameter (material length scale).However this equation(4) of strain energy also requires C^1 continuous element.

Conclusion of above both theories (2.1) and (2.2) :

In above both strain gradient theories the principle of virtual work for a linear elastic strain gradient solid can be expressed as

$$
\int_v [\sigma_{ij}\delta\varepsilon_{ij} + \tau_{ijk}\delta\eta_{ijk}]dV = \int_v [b_k\delta u_k]dV + \int_s [f_k\delta u_k + r_k D\delta u_k]dS
\tag{5}
$$

where $D(.) = n_k\frac{\partial(.)}{\partial x_k}$ is surface normal-gradient operator, b_k is the body force per unit volume of the body V while f_k and r_k are the truncation and the double stress traction per unit area of the surface S. They are in equilibrium with the Cauchy stress σ_{ij} and the higher-order stress τ_{ijk} according to

$$
b_k + (\sigma_{ij} - \tau_{jik,j})_{,i} = 0
\tag{6}
$$

The constitutive law governing the stress σ_{ij} and the higher-order stress τ_{ijk} for an elastic solid is derived through $\sigma_{ij} = \partial w/\partial\varepsilon_{ij}$ and $\tau_{ijk} = \partial w/\partial\eta_{ijk}$ where w is the strain energy density per unit volume. In this Cauchy stress σ_{ij} is work conjugate to the strain ε_{ij} and higher-order stress τ_{ijk} is work conjugate to the strain gradient η_{ijk}.

Here, second-order derivatives of displacement occurred in the principle of virtual work Eq.(5), implying that displacement-based elements of C^1-continuity are indispensable in a finite element formulation. However there are no robust C^1 continuous elements were available at that time for the application of fem formulation of above mentioned both strain gradient theories.

6

3   Different FEM Approach

There are different FEM approach to the Strain Gradient Elasticity theories are available because of the fact that C^1 continuous elements are very difficult to formulate and on the other hand there are also a mixed finite element formulation of strain gradient elasticity derived, which only requires C^0 continuity.

3.1   Tomislav Lesicar - Two scale FEM formulation

3.1.1   C1-Continuous Element

The Aifantis strain gradient theory given in subsection (2.2) has been embedded into finite element framework by Tomislav Lesicar, Zdenko Tonkovic and Jurica Soric [3]. They used the three node triangular finite element named C1PE3 [5]. The element is shown in Fig. It contains twelve degrees of freedom (DOF) per node, and it satisfies C^1 continuity with assumptions of the plane strains with unit thickness
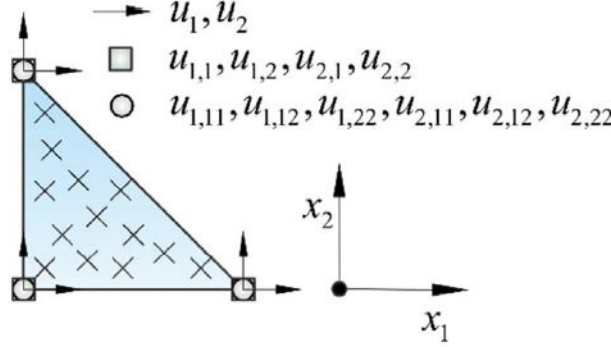
Figure 2: C1-Continuous Triangular Element

3.1.2   Higher-order stress and strain gradient tensor

They used the same weak form of the finite element formulation as given by Eq.(5). Furthermore, using the basic finite element relations strain and stress tensors can be expressed as

$$\varepsilon = Bu, \sigma = C\varepsilon \tag{7}$$

where B is a matrix containing linear combinations of the first derivatives of the components of the shape function matrix, C is an isotropic elastic constitutive matrix and u is a displacement vector. Relating to Eq.(7) the strain gradient tensor is represented as

$$\varepsilon_{,1} = \begin{bmatrix} \varepsilon_{11,1} \\ \varepsilon_{22,1} \\ 2\varepsilon_{12,1} \end{bmatrix} = B_{xx}u, \quad \varepsilon_{,2} = \begin{bmatrix} \varepsilon_{11,2} \\ \varepsilon_{22,2} \\ 2\varepsilon_{12,2} \end{bmatrix} = B_{yy}u$$

where matrices B_{xx} and B_{yy} contain linear combination of the second derivatives of the components of the shape function matrix with respect to x and y respectively. By using

above expressions the higher order stress is obtain as

$$\mu_{1ij} = \begin{bmatrix} \mu_{111} \\ \mu_{122} \\ \mu_{112} \end{bmatrix} = l^2 C\varepsilon_{,1}, \quad \mu_{2ij} = \begin{bmatrix} \mu_{211} \\ \mu_{222} \\ \mu_{212} \end{bmatrix} = l^2 C\varepsilon_{,2}$$

Finally, substituting the above expression into the virtual work Eq.(5), yields the finite element equation $Ku = F$. Here, the element stiffness matrix K is given by,

$$K = K_l + l^2(K_{xx} + K_{yy}), \tag{8}$$

where the matrices K_l, K_{xx} and K_{yy} are expressed as,

$$K_l = \int_A B^T C B dA$$
$$K_{xx} = \int_A B_{xx}^T C B_{xx} dA \tag{9}$$
$$K_{yy} = \int_A B_{yy}^T C B_{yy} dA$$

here, as observed from Eq.(8), the general stiffness matrix of the strain gradient element (C^1 continuous element) consists of the two parts, which are basic (K_l) and a higher order one ($K_{xx} + K_{yy}$). From this it can be analyse that when the microstructural length parameter l is zero this Eq.(8) is reduced to the classical one.

This element has been implemented into the FE program ABAQUS using the User Element Subroutine UEL by Tomislav Lesicar and et al. They used the reduced Gauss integration technique with 13 integration points for numerical integration of the stiffness matrices and force vector, instead of the full integration scheme with 25 points. The positions of the all 13 integration points are given in Fig. However, as discussed earlier this reduced integration technique for C^1 Continuous planar Triangular element provides not quite satisfactory results and it is more convenient for the multi scale analysis like Strain-gradient second-order computational homogenization scheme.

3.1.3 Physical interpretation of Strain Gradient

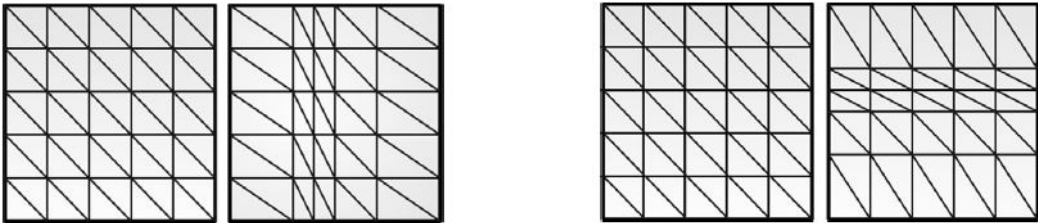Here, we can see the Physical interpretation of Strain Gradient [3].

Figure 3: Physical Interpretation of Strain Gradient η_{111} and η_{222}
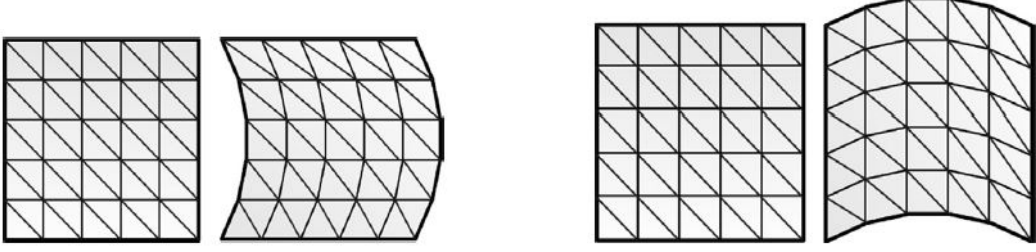
Figure 4: Physical Interpretation of Strain Gradient η_{221} and η_{112}
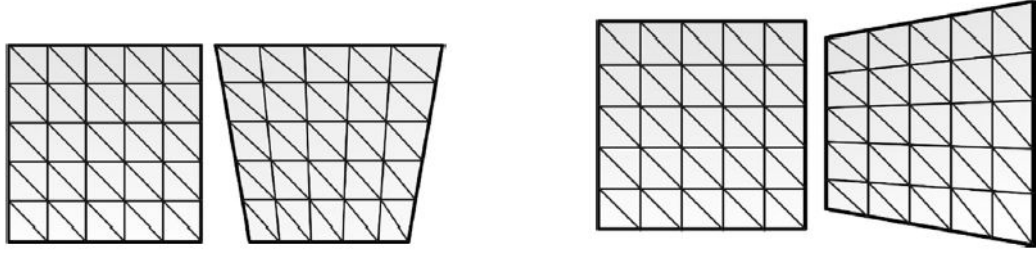
Figure 5: Physical Interpretation of Strain Gradient η_{211} and η_{122}

3.2   E. Amanatidou - Mixed type FEM formulation

In Mindlin's 1960s theory (2.1), when the problem is formulated in terms of displacements, the governing partial differential Eq.(1) is of fourth order. If traditional finite elements are used for the numerical solutions of such problems, then C^1 displacements continuity is required at inter elements.

Furthermore, E. Amanatidou [1] and et al. developed the alternative Mixed type finite element formulation, in which both displacements and the displacement gradients are used as independent unknowns and their relationship is enforced in an integral sense. In addition to that, this variational formulation can be used for both linear and nonlinear strain gradient elasticity theories. In conclusion, this finite elements requires only C^0 continuity and simple to formulate and implement into FE program using UEL.

The three equivalent forms strain energy density W given by Mindlin is expressed as,

$$W = \tilde{\mathbf{W}}(\varepsilon, \tilde{\kappa}) = \hat{\mathbf{W}}(\varepsilon, \hat{\kappa}) = \bar{\mathbf{W}}(\varepsilon, \bar{\kappa}, \bar{\bar{\kappa}}) \tag{10}$$

where the expression $W = \tilde{\mathbf{W}}(\varepsilon, \tilde{\kappa})$ known as "Type I", the expression $W = \hat{\mathbf{W}}(\varepsilon, \hat{\kappa})$ known as "Type II" and the expression $W = \bar{\mathbf{W}}(\varepsilon, \bar{\kappa}, \bar{\bar{\kappa}})$ known as "Type III".

where the Strain energy density for all three forms are represented as,

$$\tilde{\mathbf{W}}(\varepsilon, \tilde{\kappa}) = \frac{1}{2}\lambda\varepsilon_{ii}\varepsilon_{kk} + \mu\varepsilon_{ij}\varepsilon_{ij} + \frac{1}{2}l^2[\lambda\tilde{\kappa_{ijj}}\tilde{\kappa_{ikk}} + \mu(\tilde{\kappa_{ijk}}\tilde{\kappa_{ijk}} + \tilde{\kappa_{ijk}}\tilde{\kappa_{kji}})] \tag{11}$$

9

$$\hat{\mathbf{W}}(\varepsilon, \hat{\kappa}) = \frac{1}{2}\lambda\varepsilon_{ii}\varepsilon_{kk} + \mu\varepsilon_{ij}\varepsilon_{ij} + \frac{1}{2}l^2(\lambda\tilde{\kappa_{ijj}}\tilde{\kappa_{ikk}} + 2\mu\lambda\tilde{\kappa_{ijk}}\tilde{\kappa_{ijk}}) \tag{12}$$

$$\bar{\mathbf{W}}(\varepsilon, \bar{\kappa}, \bar{\bar{\kappa}}) = \frac{1}{2}\lambda\varepsilon_{ii}\varepsilon_{kk} + \mu\varepsilon_{ij}\varepsilon_{ij} + l^2[\frac{2}{9}(\lambda + 3\mu)\bar{\kappa_{ij}}\bar{\kappa_{ij}} - \frac{2}{9}\lambda\bar{\kappa_{ij}}\bar{\kappa_{ji}}$$
$$+ \frac{1}{2}\lambda\bar{\bar{\kappa_{iij}}}\bar{\bar{\kappa_{kkj}}} + \mu\bar{\bar{\kappa_{ijk}}}\bar{\bar{\kappa_{ijk}}}\frac{2}{3}\lambda e_{ijk}\bar{\kappa_{ij}}\bar{\bar{\kappa_{kpp}}}] \tag{13}$$

Amanatidou and Aravas only considered the first form of strain energy density Eq.(11), which only depends upon conventional strain ε_{ij} and higher-order strain-gradient κ_{ijk}.

3.2.1 Stress

From the Eq.(11) we can easily derived the Cauchy stress σ_{ij} as,

$$\bar{\sigma}_{ij} = \frac{\partial\tilde{\mathbf{W}}}{\partial\varepsilon_{ij}} = \lambda\varepsilon_{kk}\delta_{ij} + 2\mu\varepsilon_{ij}$$

$$= \lambda\varepsilon_{kl}\delta_{kl}\delta_{ij} + 2\mu\varepsilon_{kl}\delta_{ik}\delta_{jl} \tag{14}$$

$$= (\lambda\delta_{kl}\delta_{ij} + 2\mu\delta_{ik}\delta_{jl})\varepsilon_{kl}$$

Now, it can be easily defined the Second-order tensor C_{ijkl} using the stress-strain relation as,

$$\sigma_{ij} = C_{ijkl}\varepsilon_{kl} \tag{15}$$

By comparing the Eq.(14) and Eq.(16),

$$C_{ijkl} = \lambda\delta_{kl}\delta_{ij} + 2\mu\delta_{ik}\delta_{jl} \tag{16}$$

It can be also written in symmetric 3×3 Matrix as,

$$C = \begin{bmatrix} \lambda + 2\mu & \lambda & 0 \\ \lambda & \lambda + 2\mu & 0 \\ 0 & 0 & \frac{1}{2}\mu \end{bmatrix} \tag{17}$$

10

3.2.2 Higher Order Stress

From the Eq.(11) we can easily derived the Higher Order stress μ_{ijk} as,

$$\tilde{\mu}_{ijk} = \frac{\partial \tilde{\mathbf{W}}}{\partial \kappa_{ijk}} = \frac{1}{2}l^2[\lambda\kappa_{ijj}\delta_{ip}\delta_{kq}\delta_{kr} + \lambda\kappa_{ikk}\delta_{ip}\delta_{jq}\delta_{jr}$$

$$+ 2\mu\kappa_{ijk}\delta_{ip}\delta_{jq}\delta_{kr} + \mu\kappa_{kji}\delta_{ip}\delta_{jq}\delta_{kr} + \mu\kappa_{ijk}\delta_{kp}\delta_{jq}\delta_{ir}]$$

$$= \frac{1}{2}l^2[\lambda\kappa_{pjj}\delta_{qr} + \lambda\kappa_{pkk}\delta_{qr} + 2\mu\kappa_{pqr} + \mu\kappa_{rqq} + \mu\kappa_{rqp}]$$

$$= \frac{1}{2}l^2[\lambda\delta_{qr}(\kappa_{pjj} + \kappa_{pkk}) + 2\mu(\kappa_{pqr} + \kappa_{rqp})]$$

$$= \frac{1}{2}l^2[2\lambda\delta_{qr}\delta_{ip}\delta_{kj} + 2\mu(\delta_{ip}\delta_{jq}\delta_{kr} + \delta_{ir}\delta_{jq}\delta_{kp})]\kappa_{ijk}$$

$$= l^2[\lambda\delta_{qr}\delta_{ip}\delta_{kj} + \mu(\delta_{ip}\delta_{jq}\delta_{kr} + \delta_{ir}\delta_{jq}\delta_{kp})]\kappa_{ijk}$$

$$(18)$$

Now, it can be easily derived the Third-order tensor D_{pqrijk} using the Higher order stress and Strain gradient relation as,

$$\mu_{pqr} = D_{pqrijk}\kappa_{ijk} \tag{19}$$

By, comparing the Eq.(18) and Eq.(19),

$$D_{pqrijk} = l^2[\lambda\delta_{qr}\delta_{ip}\delta_{kj} + \mu(\delta_{ip}\delta_{jq}\delta_{kr} + \delta_{ir}\delta_{jq}\delta_{kp})] \tag{20}$$

Now, Eq.(20) can also be written in matrix notation as following, where D_{pqrijk} is a symmetric 6×6 matrix.

$$D = l^2 \begin{bmatrix} \lambda+2\mu & 0 & 0 & 0 & 0 & \frac{\lambda}{2} \\ 0 & \mu & 0 & 0 & 0 & \frac{\mu}{2} \\ 0 & 0 & \mu & 0 & \frac{\mu}{2} & 0 \\ 0 & 0 & 0 & \lambda+2\mu & \frac{\lambda}{2} & 0 \\ 0 & 0 & \frac{\mu}{2} & \frac{\lambda}{2} & \frac{\lambda+3\mu}{4} & 0 \\ \frac{\lambda}{2} & \frac{\mu}{2} & 0 & 0 & 0 & \frac{\lambda+3\mu}{4} \end{bmatrix} \tag{21}$$

So, Eq.(17) and Eq.(21) are the Voigt-notation of the constitutive equation of a general strain gradient solid introduced by Amanatidou and Aravas.
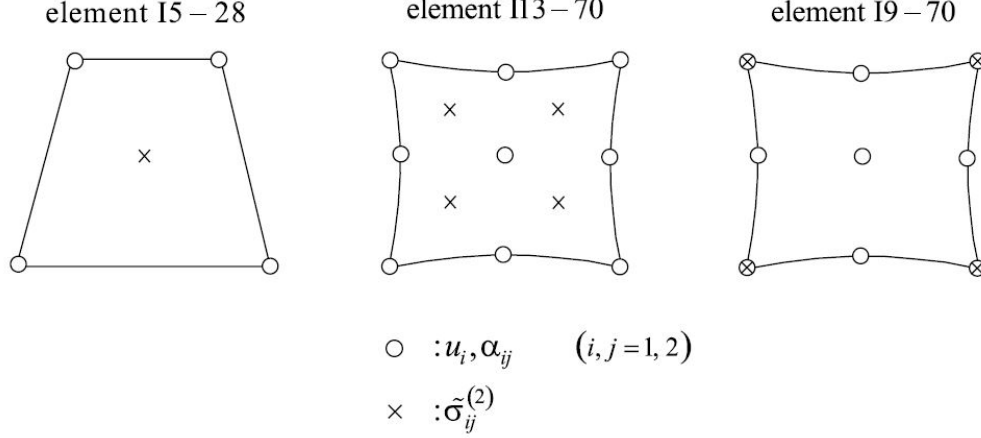
3.2.3   Mixed Tpye Elements

element I5 − 28   element I13 − 70   element I9 − 70

$\circ$ $:u_i, \alpha_{ij}$ $(i, j = 1, 2)$

$\times$ $:\tilde{\sigma}_{ij}^{(2)}$

Figure 6: C0-continuous Finite elements for Type I formulation.

Amanatidou and Aravas presented several elements that can be used in Type-I formulation are shown in above Fig. The elements are shown in Fig. , the corresponding nodal degrees of freedom are u,α and σ. Out of these given elements only the element I9-70 passes the all patch test, whereas the elements I5-28 and I13-70 failed into the patch test.

3.3   John Y. Shu - Mixed type FEM formulation

Conventional continuum mechanics theories assume that stress at a material point is a function of state variables, such as strain, at the same point. This assumption has valid until when the wavelength of a deformation field is much larger than the dominant micro-structural length scale of the material. However, when the two length scales are comparable, this assumption is proved wrong because the material behaviour at any material point is affected by the surrounding material points deformation. Therefore, FleckHutchinson strain gradient plasticity, which falls within the ToupinMindlin framework, represent the virtual work in terms of strain gradients and higher order stresses.

John Y. Shu [6] developed Mixed type FEM formulation of FleckHutchinson strain gradient elasticity theory. They devised C^0 continuous elements of mixed type, in which additional nodal degrees of freedom "Relaxed Strain" is introduced and enforce the kinematic constraints between displacement and relaxed strain by Lagrange multipliers.

3.3.1   Modified Virtual Work

Firstly, for mixed type FEM formulation, they first derive a weak form of the principle of virtual work suitable for finite element implementation using C^0-shape functions. In this modified virtual work only first-order gradients of kinematic quantities involving. In this introduced a second-order tensor ψ and a related third-order tensor η such that η_{ijk}

12

is defined as,

$$\eta_{ijk} = (\psi_{jk,i} + \psi_{ik,j})/2 \tag{22}$$

Now, modified weak form virtual work is represented as,

$$\int_{\Omega} [\sigma_{ij}\delta\varepsilon_{ij} + \tau_{ijk}\delta\eta_{ijk} + \tau_{ijk,i}(\delta\psi_{jk} - \delta u_{k,j})]d\Omega = \int_{\Omega} [b_k\delta u_k]d\Omega + \int_{\Gamma} [t_k\delta u_k + n_j r_k\delta\psi_{jk}]d\Gamma$$

$$+ \int_{\Gamma} (n_i\tau_{ijk} - n_j r_k)(\delta\psi_{jk} - \delta u_{k,j})d\Gamma \tag{23}$$

for arbitrary variations of du and dψ. If ψ is subjected strictly to the constraint of $\delta\psi = \delta u_{k,j}$ into the whole domain Ω, then as previously discussed, the strict enforcement of this constraint will demand C^1 continuous elements. Therefore, to facilitate the use of convenient C^0-continuous elements, this constraint is enforced in the following weighted residual manner as,

$$\int_{\Omega} (\psi_{jk} - u_{k,j})\delta\tau_{ijk,i}d\Omega = 0 \quad (no \quad sum \quad over \quad j \quad and \quad k) \tag{24}$$

for an arbitrary variation of the Lagrange multipliers $\tau_{ijk,i}$. Finally, by denoting the Lagrange multipliers $\delta\tau_{ijk,i}$ in above equation as $\delta\rho_{jk}$ the modified virtual work statement Eq.(23) becomes,

$$\int_{\Omega} [\sigma_{ij}\delta\varepsilon_{ij} + \tau_{ijk}\delta\eta_{ijk} + \rho_{jk}(\delta\psi_{jk} - \delta u_{k,j})]d\Omega = \int_{\Omega} [b_k\delta u_k]d\Omega + \int_{\Gamma} [t_k\delta u_k + n_j r_k\delta\psi_{jk}]d\Gamma \tag{25}$$

and the modified constraint from Eq.(24) becomes,

$$\int_{\Omega} (\psi_{jk} - u_{k,j})\delta\rho_{jk}d\Omega = 0 \quad (no \quad sum \quad over \quad j \quad and \quad k) \tag{26}$$

By considering the above criteria we get the strain energy density in terms of the relaxed strain gradient from Eq.(2) as,

$$W = \frac{1}{2}\lambda\varepsilon_{ii}\varepsilon_{jj} + \mu\varepsilon_{ij}\varepsilon_{ij} + \frac{1}{2}\mu l^2(\eta_{ijk}\eta_{ijk} - \eta_{ijk}\eta_{kji}) \tag{27}$$

3.3.2 Stress

From the Eq.(27) we can easily derived the Cauchy stress σ_{ij} as,

$$\bar{\sigma}_{ij} = \frac{\partial \tilde{\mathbf{W}}}{\partial\varepsilon_{ij}} = \lambda\varepsilon_{kk}\delta_{ij} + 2\mu\varepsilon_{ij} = (\lambda\delta_{kl}\delta_{ij} + 2\mu\delta_{ik}\delta_{jl})\varepsilon_{kl} = C_{ijkl}\varepsilon_{kl} \tag{28}$$

So, it yields the same result as given by Eq.(14) and Eq.(16). Therefore, we can also write the matrix notation of C_{ijkl} and it also same as Eq.(17).

13

3.3.3   Higher-order Stress

From the Eq.(27) we can easily derived the Higher-order stress τ_{ijk} as,

$$
\begin{aligned}
\tau_{pqr} &= \frac{\partial \tilde{\mathbf{W}}}{\partial \eta_{pqr}} \\
&= \frac{1}{2}\mu l^2 [2(\eta_{ijk}\delta_{ip}\delta_{jq}\delta_{kr}) - \eta_{kji}\delta_{ip}\delta_{jq}\delta_{kr} - \eta_{ijk}\delta_{kp}\delta_{jq}\delta_{ir}] \\
&= \frac{1}{2}\mu l^2 [2(\eta_{pqr}) - \eta_{rqp} - \eta_{rqp}] \\
&= \mu l^2 [\eta_{pqr} - \eta_{rqp}] \\
&= \mu l^2 [\delta_{lp}\delta_{mq}\delta_{nr} - \delta_{lr}\delta_{mq}\delta_{np}]\eta_{lmn}
\end{aligned}
\tag{29}
$$

Now, it can be easily derived the Third-order tensor D_{pqrlmn} using the Higher order stress τ_{pqr} and Strain gradient η_{lmn} relation as,

$$
\tau_{pqr} = D_{pqrlmn} \cdot \eta_{lmn}
\tag{30}
$$

By, comparing the Eq.(29) and Eq.(30),

$$
D_{pqrlmn} = \mu l^2 [\delta_{lp}\delta_{mq}\delta_{nr} - \delta_{lr}\delta_{mq}\delta_{np}]
\tag{31}
$$

Now, Eq.(29) can also be written in matrix notation as following, where D_{pqrlmn} is a symmetric 6×6 matrix.

$$
D = \mu l^2
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & -\frac{1}{2} \\
0 & 0 & 1 & 0 & -\frac{1}{2} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -\frac{1}{2} & 0 & \frac{1}{4} & 0 \\
0 & -\frac{1}{2} & 0 & 0 & 0 & \frac{1}{4}
\end{bmatrix}
\tag{32}
$$

So, Eq.(17) and Eq.(32) are the Voigt-notation of the constitutive equation of a Couple strain gradient solid represented by JOHN Y. SHU, WAYNE E. KING and NORMAN A. FLECK.

3.3.4   Isoparametric elements for the strain gradient solid

DOF at ■ : u_1, u_2, Ψ_{11}, Ψ_{21}, Ψ_{12}, Ψ_{22}

DOF at ● : u_1, u_2

Lagrange multipliers at × : ρ_{11}, ρ_{21}, ρ_{12}, ρ_{22}

Lagrange multiplier (for incompressible solid) at ○: σ_m
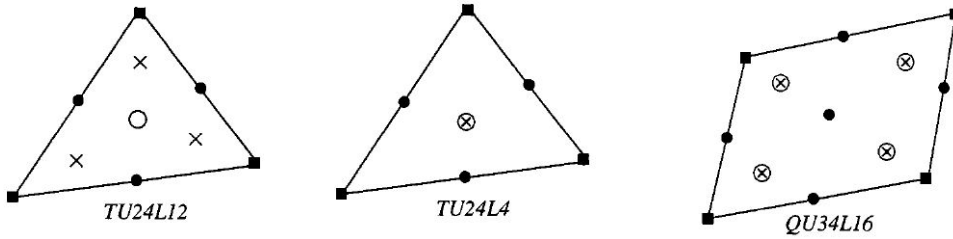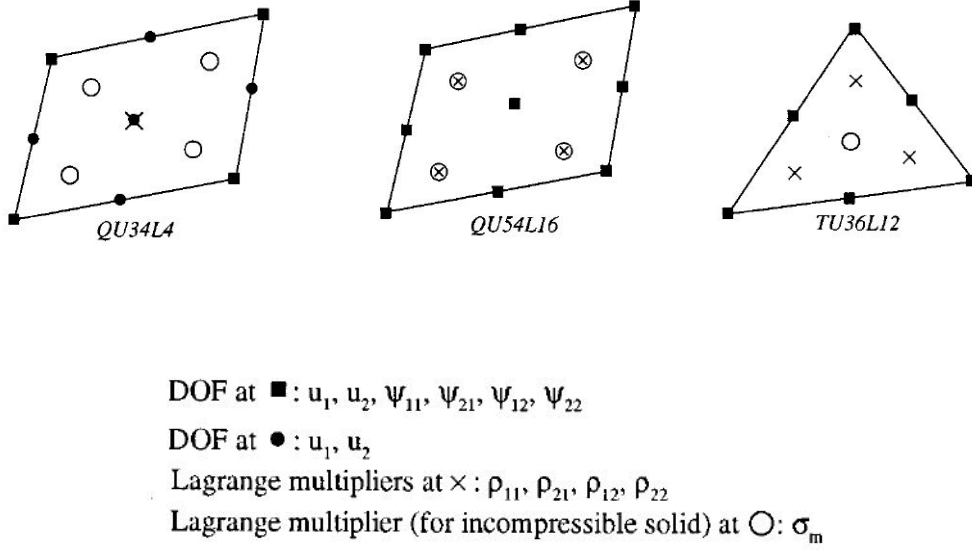
Figure 7: C0-continuous 6 Finite elements for Mixed type of formulation.

A total of six types of triangular element and quadrilateral element have been developed by JOHN Y. SHU for strain gradient theory. In which Displacement gradients are introduced as extra nodal degrees of freedom in terms of relaxed strain. The kinematic constraints between the relaxed strain and true gradients of displacement are enforced via Lagrange multipliers. All six types of element passed a patch test. However, he recommended QU34L4 elements for practical applications, since it gives precise accuracy than other elements.

4 FEM Implementation

In subsection "John Y. Shu - Mixed type FEM formulation", there are 6 different triangular element and quadrilateral element represented out of which element QU34L4 is recommended for practical applications. so here we will use the QU34L4 element in order to implement strain gradient theory.

4.1 QU34L4 - ELEMENT

This QU34L4 is Shown in below Fig. ,which contains total 9 nodes and 38 degrees of freedom. In this element there are two displacements DOF u_1 and u_2 at each nodes of it. There are newly introduced relaxed strain DOF $\psi_{11}, \psi_{21}, \psi_{12}$ and ψ_{22} at each corner nodes of it. In addition to that there are also four Lagrange multipliers ρ_{11}, ρ_{21}, ρ_{12} and ρ_{22}, which are assumed to be constant throughout the element. So, in QU34L4 Q stands for nine-noded isoparametric quadrilateral element, where U stands for number of DOF related to the displacement which are total 34 in this element, where L stands for Lagrange multipliers which are total 4 in this element.
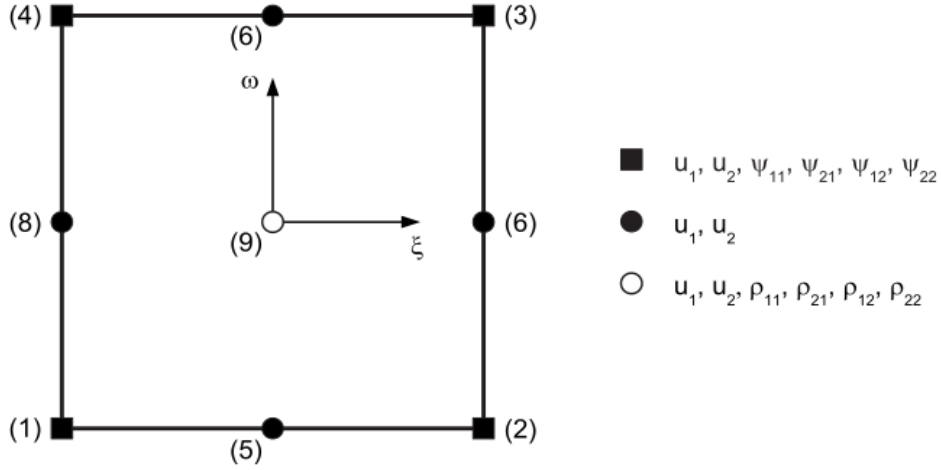

Figure 8: C0-continuous QU34L4 Element.

Displacements u_1, u_2 are interpolated using standard quadratic shape functions in terms of the area co-ordinates. Relaxed displacement gradients $\psi_{11}, \psi_{21}, \psi_{12}$ and ψ_{22} are interpolated linearly. The different interpolation is motivated by the consideration that the true displacement gradients ε_{ij} will depend linearly on the co-ordinates if all sides of the element are straight.

4.2 Degrees of Freedom(DOF)

In this section we derive the finite element shape function and its derivatives for the different DOF - displacements and relaxed strain. In this we also defined matrices for langrange multipliers and also established the matrices containing the shape functions and its derivatives.

16

4.2.1 Displacement DOF

As discussed in above section, there are two displacements DOF u_1 and u_2 at each nodes. From the weak of modified virtual work we can derive the relation between global displacements and nodal-displacement as,

$$u^{(e)} = \begin{bmatrix} u_1^{(e)} \\ u_2^{(e)} \end{bmatrix} = N^{[u]\mathbf{T}} \cdot \hat{u}^{(e)} \tag{33}$$

where $\hat{u}_{(e)}$ is defined in terms of vector as,

$$\hat{u}^{(e)} = \begin{bmatrix} u_1^{(1)} & u_2^{(1)} & u_1^{(2)} & u_2^{(2)} & \cdots & \cdots u_1^{(9)} u_2^{(9)} \end{bmatrix} \tag{34}$$

In QU34L4 element, as discussed earlier u_i are interpolated using standard quadratic shape functions in terms of the area co-ordinates. So here we derived the shape functions for nine-nodded element in coordinates ξ and ω.

$$
\begin{aligned}
N^{[U]^{(1)}} &= \frac{1}{4}\xi(\xi - 1)\omega(\omega - 1) \\
N^{[U]^{(2)}} &= \frac{1}{4}\xi(\xi + 1)\omega(\omega - 1) \\
N^{[U]^{(3)}} &= \frac{1}{4}\xi(\xi + 1)\omega(\omega + 1) \\
N^{[U]^{(4)}} &= \frac{1}{4}\xi(\xi - 1)\omega(\omega + 1) \\
N^{[U]^{(5)}} &= \frac{1}{2}(1 - \xi^2)\omega(\omega - 1) \\
N^{[U]^{(6)}} &= \frac{1}{2}\xi(\xi + 1)(1 - \omega^2) \\
N^{[U]^{(7)}} &= \frac{1}{2}(1 - \xi^2)\omega(\omega + 1) \\
N^{[U]^{(8)}} &= \frac{1}{2}\xi(\xi - 1)(1 - \omega^2) \\
N^{[U]^{(9)}} &= (1 - \xi^2)(1 - \omega^2)
\end{aligned} \tag{35}
$$

From equations Eq.(33), Eq.(34) and Eq.(35) we can assemble the shape matrix as following,

$$N^{[u]T} = \begin{bmatrix} N^{[u]^{(1)}} & 0 & N^{[u]^{(2)}} & 0 & \cdots & \cdots & N^{[u]^{(9)}} & 0 \\ 0 & N^{[u]^{(1)}} & 0 & N^{[u]^{(2)}} & \cdots & \cdots & 0 & N^{[u]^{(9)}} \end{bmatrix} \tag{36}$$

Now we have shape functions, so in next step we derive the local derivatives of the shape functions with respect to ξ and ω.

The local derivatives of shape function with respect to are coordinates ξ and ω is given as,

$$N_{,\xi}^{[U]^{(i)}} = \frac{\partial N^{[U]^{(i)}}}{\partial \xi} \tag{37}$$

$$N_{,\omega}^{[U]^{(i)}} = \frac{\partial N^{[U]^{(i)}}}{\partial \omega} \tag{38}$$

From Eq.(37) we can get the local derivatives w.r.t ξ as,

$$
\begin{aligned}
N_{,\xi}^{[U]^{(1)}} &= -\frac{1}{2}\xi\omega - \frac{1}{4}\omega^2 + \frac{1}{4}\omega + \frac{1}{2}\xi\omega^2 \\
N_{,\xi}^{[U]^{(2)}} &= -\frac{1}{2}\xi\omega + \frac{1}{4}\omega^2 - \frac{1}{4}\omega + \frac{1}{2}\xi\omega^2 \\
N_{,\xi}^{[U]^{(3)}} &= \frac{1}{2}\xi\omega + \frac{1}{4}\omega^2 + \frac{1}{4}\omega + \frac{1}{2}\xi\omega^2 \\
N_{,\xi}^{[U]^{(4)}} &= \frac{1}{2}\xi\omega - \frac{1}{4}\omega^2 - \frac{1}{4}\omega + \frac{1}{2}\xi\omega^2 \\
N_{,\xi}^{[U]^{(5)}} &= \xi\omega - \xi\omega^2 \\
N_{,\xi}^{[U]^{(6)}} &= \xi - \xi\omega^2 + \frac{1}{2} - \frac{1}{2}\omega^2 \\
N_{,\xi}^{[U]^{(7)}} &= -\xi\omega - \xi\omega^2 \\
N_{,\xi}^{[U]^{(8)}} &= \xi - \xi\omega^2 - \frac{1}{2} + \frac{1}{2}\omega^2 \\
N_{,\xi}^{[U]^{(9)}} &= -2\xi + 2\xi\omega^2
\end{aligned}
\tag{39}
$$

From Eq.(38) we can get the local derivatives w.r.t ω as,

$$
\begin{aligned}
N_{,\omega}^{[U]^{(1)}} &= -\frac{1}{2}\xi\omega - \frac{1}{4}\xi^2 + \frac{1}{4}\xi + \frac{1}{2}\xi^2\omega \\
N_{,\omega}^{[U]^{(2)}} &= \frac{1}{2}\xi\omega - \frac{1}{4}\xi^2 - \frac{1}{4}\xi + \frac{1}{2}\xi^2\omega \\
N_{,\omega}^{[U]^{(3)}} &= \frac{1}{2}\xi\omega + \frac{1}{4}\xi^2 + \frac{1}{4}\xi + \frac{1}{2}\xi^2\omega \\
N_{,\omega}^{[U]^{(4)}} &= -\frac{1}{2}\xi\omega + \frac{1}{4}\xi^2 - \frac{1}{4}\xi + \frac{1}{2}\xi^2\omega \\
N_{,\omega}^{[U]^{(5)}} &= \frac{1}{2}\xi^2 - \frac{1}{2} - \xi^2\omega + \omega \\
N_{,\omega}^{[U]^{(6)}} &= -\xi^2\omega - \xi\omega \\
N_{,\omega}^{[U]^{(7)}} &= -\frac{1}{2}\xi^2 + \frac{1}{2} - \xi^2\omega + \omega \\
N_{,\omega}^{[U]^{(8)}} &= -\xi^2\omega + \xi\omega \\
N_{,\omega}^{[U]^{(9)}} &= -2\omega + 2\xi^2\omega
\end{aligned}
\tag{40}
$$

18

Taking the equations above into consideration, the displacement gradient in the element is derived as following,

$$\nabla u^{(e)} = \begin{Bmatrix} u_{1,1}^{(e)} \\ u_{1,2}^{(e)} \\ u_{2,1}^{(e)} \\ u_{2,2}^{(e)} \end{Bmatrix} = M^{[u]\mathbf{T}} u^{(e)} \tag{41}$$

So here in Eq.(41), the $u_{1,2}$ and $u_{2,1}$ are not same as in consideration of normal strain, where we considered $\varepsilon_{1,2}$ and $\varepsilon_{2,1}$ are same.

From Eq.(34) and Eq.(41), we can defined the Displacements Gradient Matrix as following,

$$M^{[u]\mathbf{T}} = \begin{bmatrix} N_{,1}^{[u](1)} & 0 & N_{,1}^{[u](2)} & 0 & \cdots & \cdots & N_{,1}^{[u](9)} & 0 \\ 0 & N_{,2}^{[u](1)} & 0 & N_{,2}^{[u](2)} & \cdots & \cdots & 0 & N_{,2}^{[u](9)} \\ N_{,1}^{[u](1)} & 0 & N_{,1}^{[u](2)} & 0 & \cdots & \cdots & N_{,1}^{[u](9)} & 0 \\ 0 & N_{,2}^{[u](1)} & 0 & N_{,2}^{[u](2)} & \cdots & \cdots & 0 & N_{,2}^{[u](9)} \end{bmatrix} \tag{42}$$

Now, we can derived the true strain matrix as following,

$$\varepsilon^{(e)} = \begin{Bmatrix} \varepsilon_{11}^{(e)} \\ \varepsilon_{22}^{(e)} \\ \gamma_{12}^{(e)} \end{Bmatrix} \approx B^{[u]\mathbf{T}} u^{(e)} \tag{43}$$

From Eq.(34) and Eq.(43), we can defined the differential Matrix B^u, which contains the global derivatives of shape functions as following,

$$B^{[u]\mathbf{T}} = \begin{bmatrix} N_{,1}^{[u](1)} & 0 & N_{,1}^{[u](2)} & 0 & \cdots & \cdots & N_{,1}^{[u](9)} & 0 \\ 0 & N_{,2}^{[u](1)} & 0 & N_{,2}^{[u](2)} & \cdots & \cdots & 0 & N_{,2}^{[u](9)} \\ N_{,2}^{[u](1)} & N_{,1}^{[u](1)} & N_{,2}^{[u](2)} & N_{,1}^{[u](2)} & \cdots & \cdots & N_{,2}^{[u](9)} & N_{,1}^{[u](9)} \end{bmatrix} \tag{44}$$

In both the Eq.(42) and Eq.(44), it can be seen that there $N_{,1}$ and $N_{,2}$ which are represented as $N_{,1}^{[u](i)} = \dfrac{\partial N^{[u](i)}}{\partial x_1}$ and $N_{,2}^{[u](i)} = \dfrac{\partial N^{[u](i)}}{\partial x_2}$, which are global derivatives of shape functions related to displacements DOF. which can not be calculated directly since the shape functions are set up in the local coordinate system.

4.2.2 Relaxed Strain DOF

As discussed in above section, there are four relaxed strain DOF $\psi_{11}, \psi_{21}, \psi_{12}$ and ψ_{22}, at each corned nodes. Now, we can derive the relation between global relaxed strain and nodal-relaxed strain as,

$$
\psi^{(e)} = \begin{Bmatrix} \psi_{11}^{(e)} \\ \psi_{21}^{(e)} \\ \psi_{12}^{(e)} \\ \psi_{22}^{(e)} \end{Bmatrix} = N^{[\psi]^{\mathbf{T}}} \hat{\psi}^{(e)} \tag{45}
$$

where, nodal relaxed strain vector $\hat{\psi}^{(e)}$ is defined as following,

$$
\hat{\psi}^{(e)\mathbf{T}} = \begin{bmatrix} \psi_{(11)}^{(1)} & \psi_{(21)}^{(1)} & \psi_{(12)}^{(1)} & \psi_{(22)}^{(1)} & \cdots & \cdots & \psi_{(11)}^{(4)} & \psi_{(21)}^{(4)} & \psi_{(12)}^{(4)} & \psi_{(22)}^{(4)} \end{bmatrix} \tag{46}
$$

Now, defined the shape functions for the relaxed strain. As discussed earlier, in QU34L4 element ψ_i are interpolated using standard bi-linear shape functions in terms of the area coordinates ξ and ω.

$$
\begin{aligned}
N^{[\psi]^{(1)}} &= \frac{1}{4}(1-\xi)(1-\omega) \\
N^{[\psi]^{(2)}} &= \frac{1}{4}(1+\xi)(1-\omega) \\
N^{[\psi]^{(3)}} &= \frac{1}{4}(1+\xi)(1+\omega) \\
N^{[\psi]^{(4)}} &= \frac{1}{4}(1-\xi)(1+\omega)
\end{aligned} \tag{47}
$$

Now, considering the equations Eq.(45), Eq.(46) and Eq.(47) we can assemble the shape matrix for the relaxed strain degree of freedom as following,

$$
N^{[\psi]^{\mathbf{T}}} = \begin{bmatrix} N^{\psi^{(1)}} & 0 & 0 & 0 & \cdots & \cdots & N^{\psi^{(4)}} & 0 & 0 & 0 \\ 0 & N^{\psi^{(1)}} & 0 & 0 & \cdots & \cdots & 0 & N^{\psi^{(4)}} & 0 & 0 \\ 0 & 0 & N^{\psi^{(1)}} & 0 & \cdots & \cdots & 0 & 0 & N^{\psi^{(4)}} & 0 \\ 0 & 0 & 0 & N^{\psi^{(1)}} & \cdots & \cdots & 0 & 0 & 0 & N^{\psi^{(4)}} \end{bmatrix} \tag{48}
$$

Now we have shape functions for relaxed strain DOF Eq.(47), so in next step we derive the local derivatives of the shape functions with respect to ξ and ω.
The local derivatives of shape function with respect to are coordinates ξ and ω is given as,

$$
N_{,\xi}^{[\psi]^{(i)}} = \frac{\partial N^{[\psi]^{(i)}}}{\partial \xi} \tag{49}
$$

$$N_{,\omega}^{[\psi](i)} = \frac{\partial N^{[\psi](i)}}{\partial \omega} \tag{50}$$

From Eq.(49) we can get the local derivatives w.r.t ξ as,

$$
\begin{aligned}
N_{,\xi}^{[\psi](1)} &= \frac{1}{4}(-1 + \omega) \\
N_{,\xi}^{[\psi](2)} &= \frac{1}{4}(1 - \omega) \\
N_{,\xi}^{[\psi](3)} &= \frac{1}{4}(1 + \omega) \\
N_{,\xi}^{[\psi](4)} &= \frac{1}{4}(-1 - \omega)
\end{aligned}
\tag{51}
$$

From Eq.(50) we can get the local derivatives w.r.t ω as,

$$
\begin{aligned}
N_{,\omega}^{[\psi](1)} &= \frac{1}{4}(-1 + \xi) \\
N_{,\omega}^{[\psi](2)} &= \frac{1}{4}(-1 - \xi) \\
N_{,\omega}^{[\psi](3)} &= \frac{1}{4}(1 + \xi) \\
N_{,\omega}^{[\psi](4)} &= \frac{1}{4}(1 - \xi)
\end{aligned}
\tag{52}
$$

Now, we defined the strain gradient using the differential matrix Eq.(54) and nodal relaxed strain Eq.(46)

$$
\eta^{(e)} = \begin{Bmatrix} \eta_{111}^{(e)} \\[6pt] \eta_{221}^{(e)} \\[6pt] \eta_{112}^{(e)} \\[6pt] \eta_{222}^{(e)} \\[6pt] \phi_1^{(e)} \\[6pt] \phi_2^{(e)} \end{Bmatrix} \approx B^{[\psi]\mathbf{T}} \psi^{(e)} \tag{53}
$$

Form, considering Eq.(46) and Eq.(53) now we can defined the differential matrix B^ψ as following,

$$
B^{[\psi]\mathbf{T}} = \begin{bmatrix}
N_{,1}^{[\psi](1)} & 0 & 0 & 0 & \cdots & \cdots & N_{,1}^{[\psi](4)} & 0 & 0 & 0 \\
0 & N_{,2}^{[\psi](1)} & 0 & 0 & \cdots & \cdots & 0 & N_{,2}^{[\psi](4)} & 0 & 0 \\
0 & 0 & N_{,1}^{[\psi](1)} & 0 & \cdots & \cdots & 0 & 0 & N_{,1}^{[\psi](4)} & 0 \\
0 & 0 & 0 & N_{,2}^{[\psi](1)} & \cdots & \cdots & 0 & 0 & 0 & N_{,2}^{[\psi](4)} \\
N_{,2}^{[\psi](1)} & N_{,1}^{[\psi](1)} & 0 & 0 & \cdots & \cdots & N_{,2}^{[\psi](4)} & N_{,1}^{[\psi](4)} & 0 & 0 \\
0 & 0 & N_{,2}^{[\psi](1)} & N_{,1}^{[\psi](1)} & \cdots & \cdots & 0 & 0 & N_{,2}^{[\psi](4)} & N_{,1}^{[\psi](4)}
\end{bmatrix}
\tag{54}
$$

In this above Eq.(54), it can be seen that there $N_{,1}^{\psi}$ and $N_{,2}^{\psi}$ which are represented as $N_{,1}^{[\psi]^{(i)}} = \dfrac{\partial N^{[\psi]^{(i)}}}{\partial x_1}$ and $N_{,2}^{[\psi]^{(i)}} = \dfrac{\partial N^{[\psi]^{(i)}}}{\partial x_2}$, which are global derivatives of shape functions related to displacements DOF. which can not be calculated directly since the shape functions are set up in the local coordinate system ξ and ω.

4.2.3 Langrange Multiplier

The Lagrange multipliers in the element $\rho^{(e)}$ are defined as,

$$
\rho^{(e)} = \begin{Bmatrix} \rho_{11}^{(e)} \\ \rho_{21}^{(e)} \\ \rho_{12}^{(e)} \\ \rho_{22}^{(e)} \end{Bmatrix} = N^{[\rho]^{\mathbf{T}}} \rho^{(e)} \tag{55}
$$

As discussed earlier in above section, it is assumed that Lagrange multipliers constant throughout the element they are only existent at the middle node and therefore, the shape functions matrix for it defined as,

$$
N^{[\rho]} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{56}
$$

5 Verification of model without Gradient term

After completely programmed the Strain Gradient theory for FEM model into Fortran programming language. I tried to run this code with ABAQUS. And Surprisingly, I got the different errors, out of which main errors are Convergence errors, NAN error and singularities error. In conclusion of that I have decided to debug my code from the scratch. Therefore, I have decided to check for the simpler case which is Mechanical Displacement degree of freedom. So first i wanted to check that my code is correctly running for the Displacement degree if freedom.

So, I have just commented all the terms which is related to gradient term i.e. which is related to the relaxed strain and Lagrange degree of freedom. After that I have implement my code to the plate-720 geometry which is explained in detailed in later sections.

Now, I have implement the my code which is correctly running after the removing of many hurdles in terms of programming error. Here, I representing the my code results for simpler loading conditions and compare it with the ABAQUS results.

So, here I represent the displacement results of ABAQUS with simpler loading i.e. 1% of displacement in positive Y direction with my results without including the gradient terms and any other degrees of freedom.
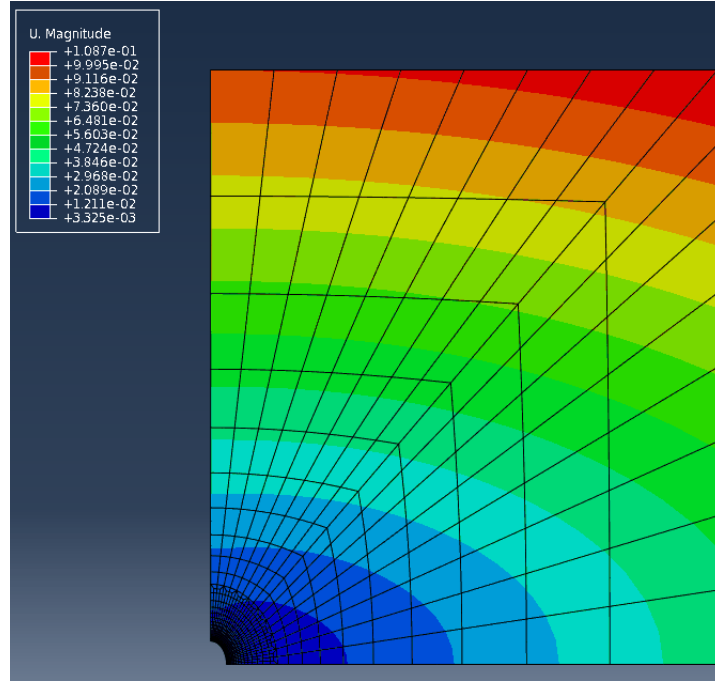

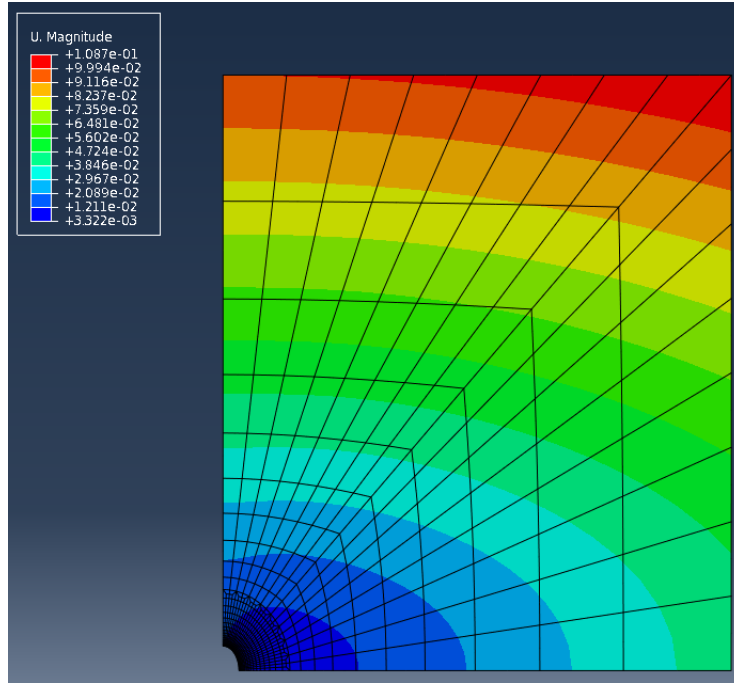Figure 9: Displacement visualization of ABAQUS Model.

Figure 10: Displacement visualization of Strain Gradient Model.

So, here I represent the Stress results of ABAQUS with simpler loading i.e. 1% of displacement in positive Y direction with my results without including the gradient terms and any other degrees of freedom. In the second fig I have represent the complex part of geometry.
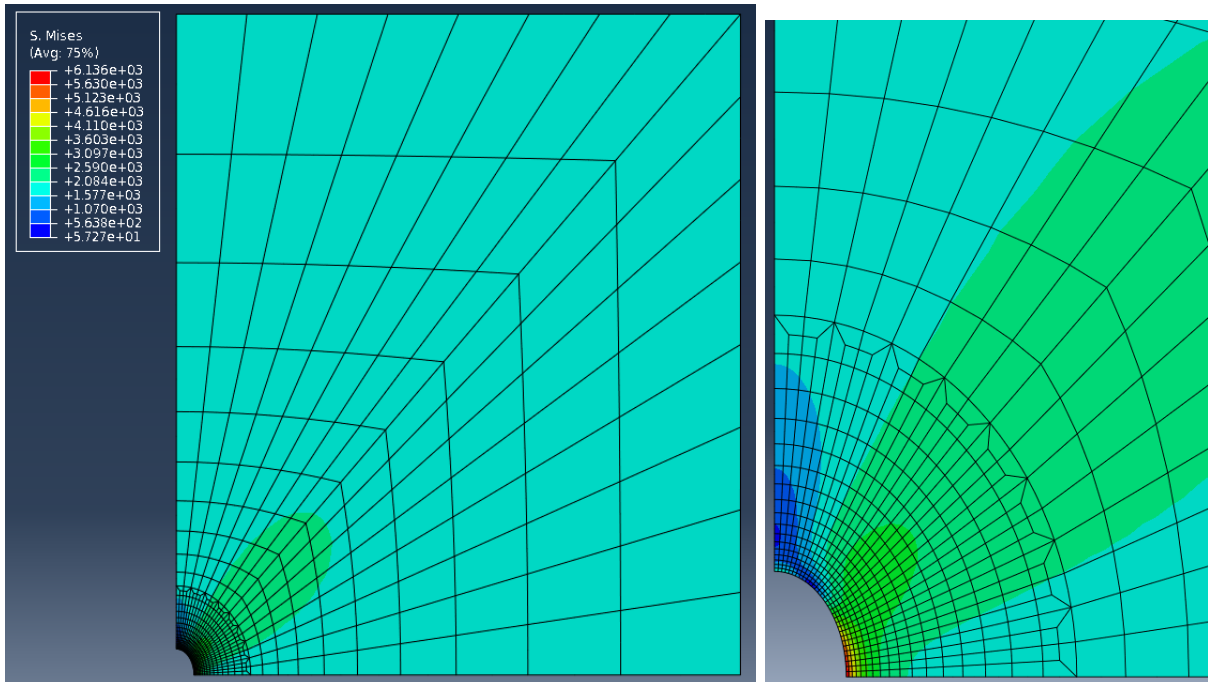

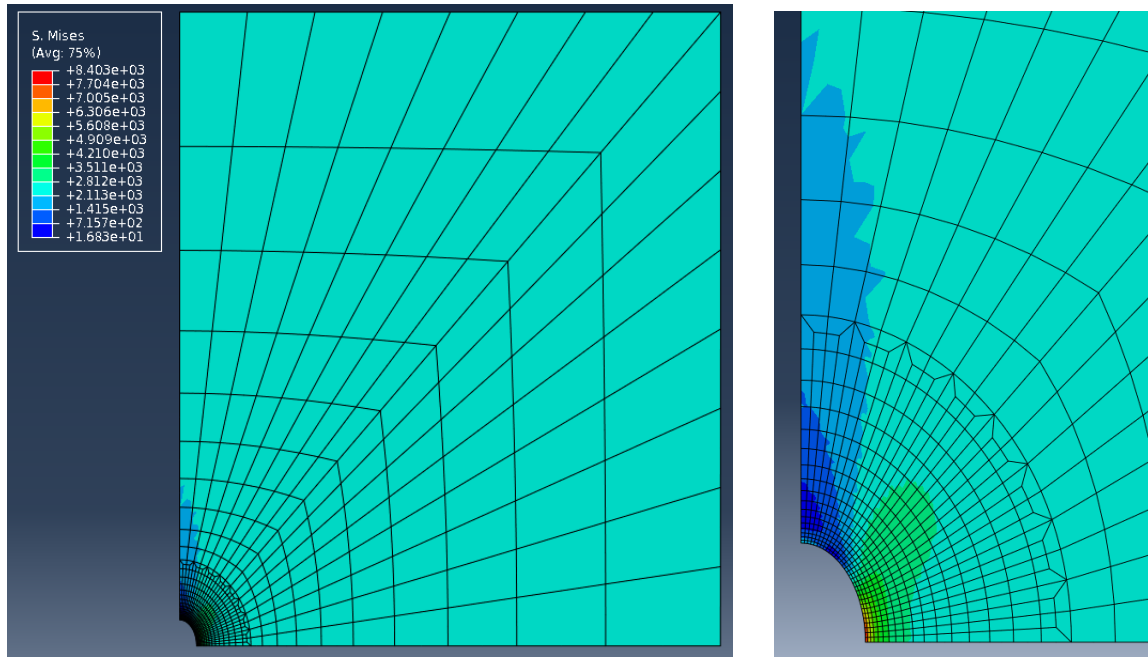Figure 11: Stress visualization of ABAQUS Model.

24

Figure 12: Stress visualization of Strain Gradient Model.

6   Verification of Model Plate-720

As of described in E. Amanatidou - Mixed type FEM formulation, they have used the PLATE-720 which contains the 720 elements and 2983 nodes. Using this geometry they try to find the stress concentration factor and plot the stress concentration factor for different microstructural length.

So, here I have used the same geometry i.e. Plate with domain length of 10 and hole radius is 0.333. I also mesh it as same as described in that paper with also transition of fine mesh into the coarse mesh, which is at 1.332 from the centre of the hole.
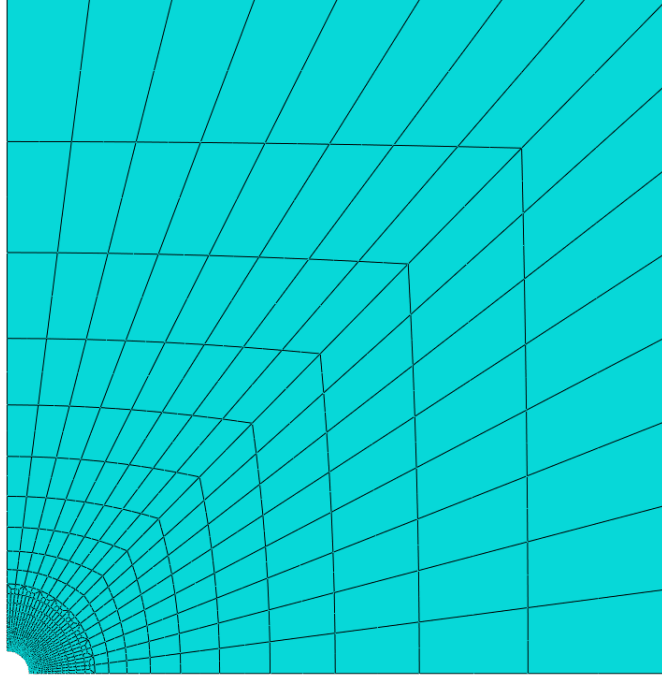

Figure 13: Meshed Plate-720 with hole in the left-bottom corner.

Here, I specifically represent the fine mesh around the hole and also the transition of mesh from coarse mesh to fine mesh at specific position from the centre of the hole.
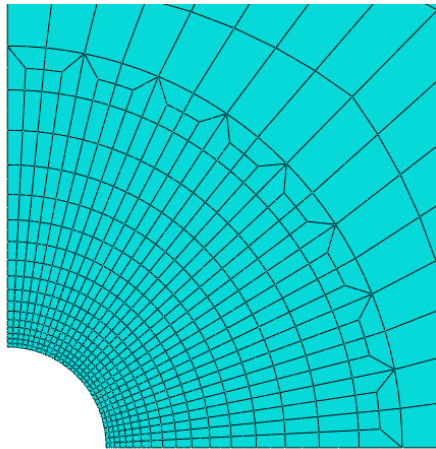

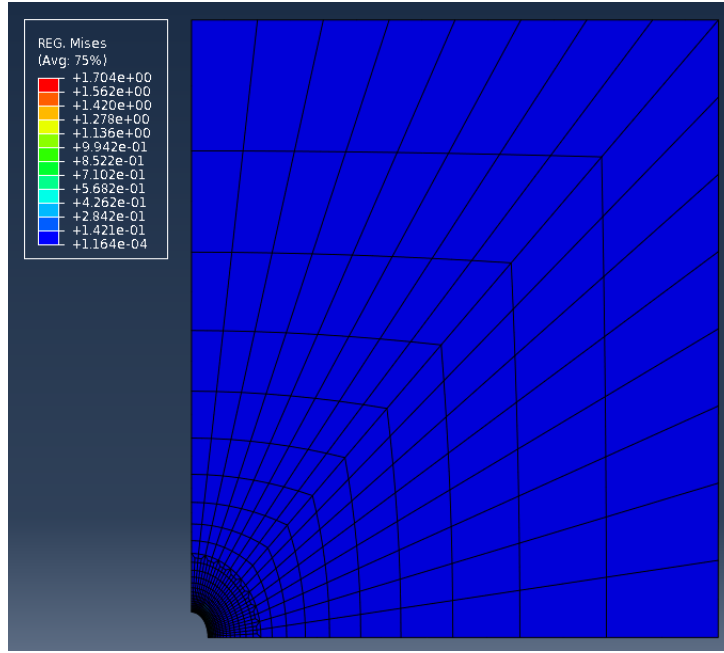Figure 14: Fine mesh and Transition in Plate-720.

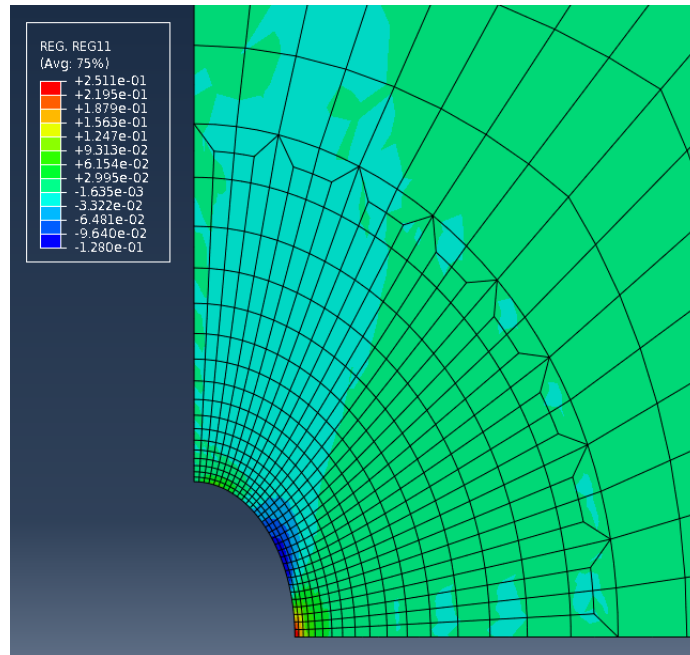Figure 15: Strain Gradient Visualization in Plate-720.

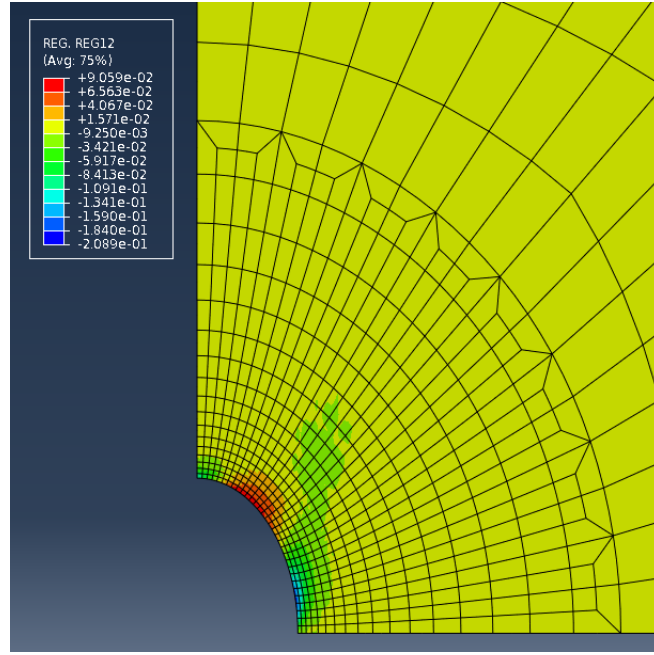Figure 16: Visualization of Strain Gradient η_{111}

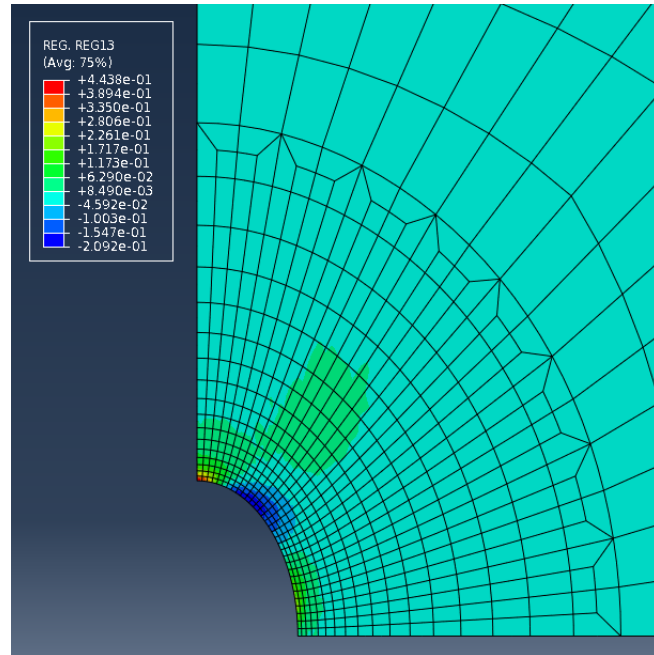Figure 17: Visualization of Strain Gradient η_{221}

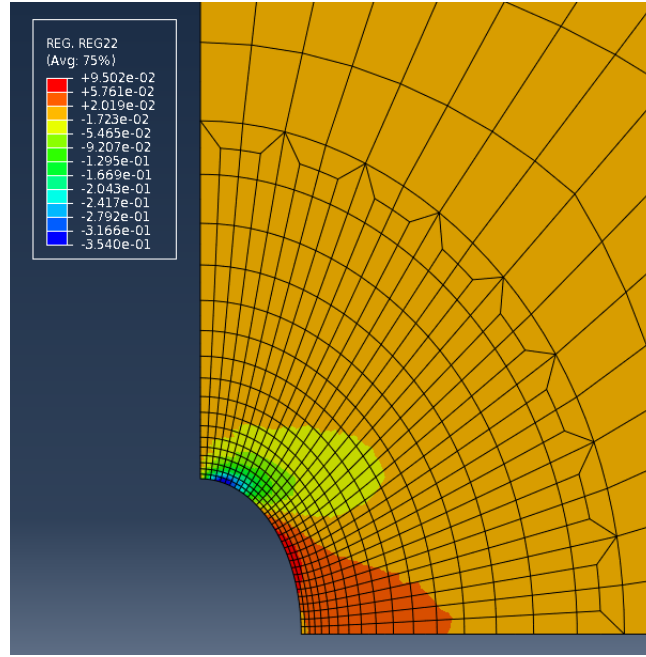Figure 18: Visualization of Strain Gradient η_{112}

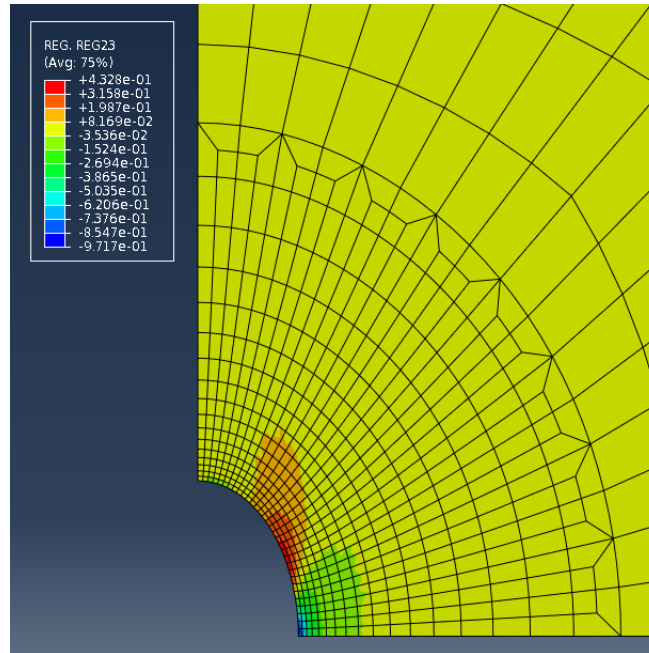Figure 19: Visualization of Strain Gradient η_{222}

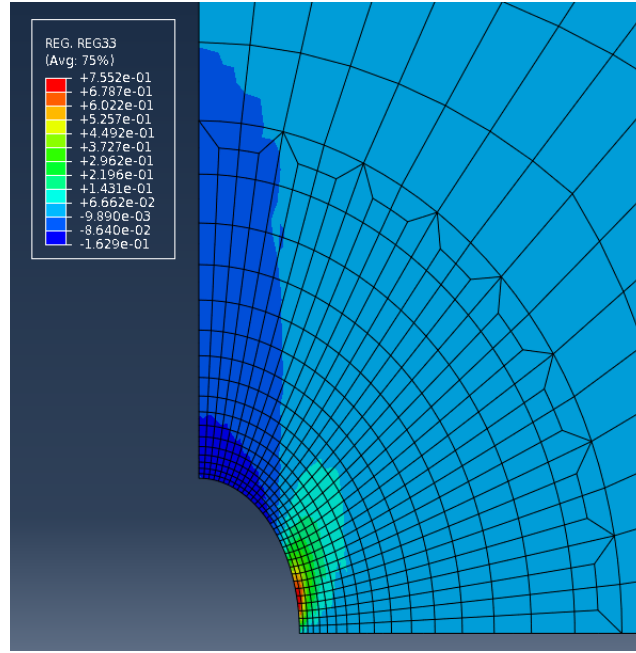Figure 20: Visualization of Strain Gradient $\eta_{121} + \eta_{211}$

Figure 21: Visualization of Strain Gradient $\eta_{122} + \eta_{212}$

7 Patch Tests

In this chapter I present the Patch test which is implemented with the one element and four element of same geometry using finite element method. The Patch tests are necessary for checking the stability of the newly developed user element and also important to check the convergence of the finite element solutions.

7.1 Patch Test - 1

Now, we considered an one Square geometry with specific dimensions as given below. In below given fig L = 1.0 and so therefore the coordinates of the points A(0,0), B(1,0), C(1,1) and D(0,1).
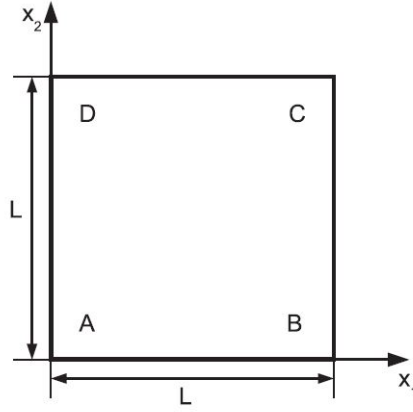

Figure 22: Square Domain with one element for implementation of Patch Test-1.

In this Patch test we just find the analytical solution for displacements and relaxed strain. After that we are finding the values of the displacements and relaxed strain at each nodes and each corner nodes respectively by assigning the position of that node. In the last we implement the all nodal degrees of freedom in terms of the boundary condition and we find the stiffness matrix and right hand side matrix for one element by running the code with ABAQUS and we try to prove the equilibrium equation.

In details, we have total 18 displacements values of all nodes and 16 relaxed strain values of each corner nodes of the element which are prescribe in the Input file for one element in terms of boundary condition.

For proving the equilibrium equation, we have stiffness matrix, RHS matrix and displacement matrix in hand. But the dimension of the above matrices are big so we wouldn't prefer to solve it on paper by hand. so i write one python scripts which reads the data of all matrices from individual files. So for that now we have to put the stiffness matrix into 'AMATRX.dat', RHS matrix into 'RHS.dat' and displacement matrix into 'U.dat' file respectively. This python file read the data and store it in terms of multidimensional array and that try to find the residual of the equilibrium equation and print it into terminal so we can verify our results.
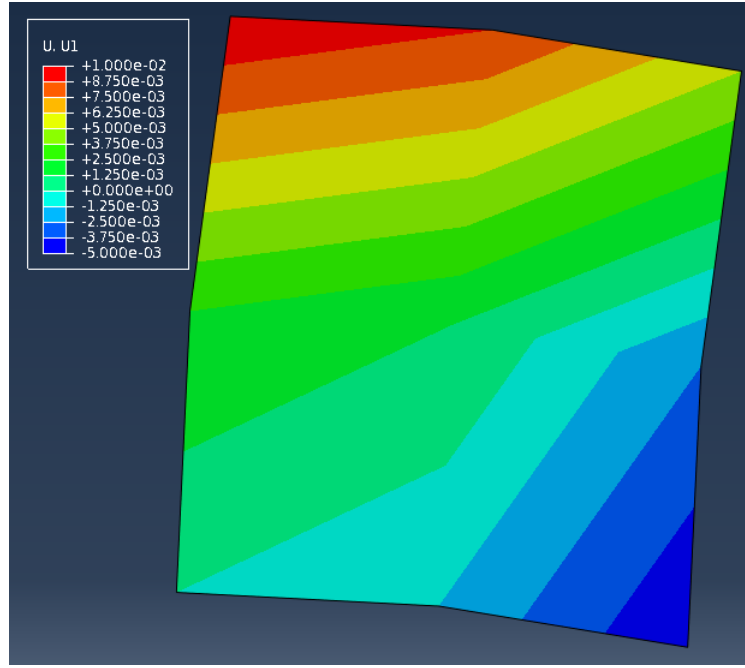
Figure 23: Visualization of Displacement u_1 in Patch Test-1 Model.
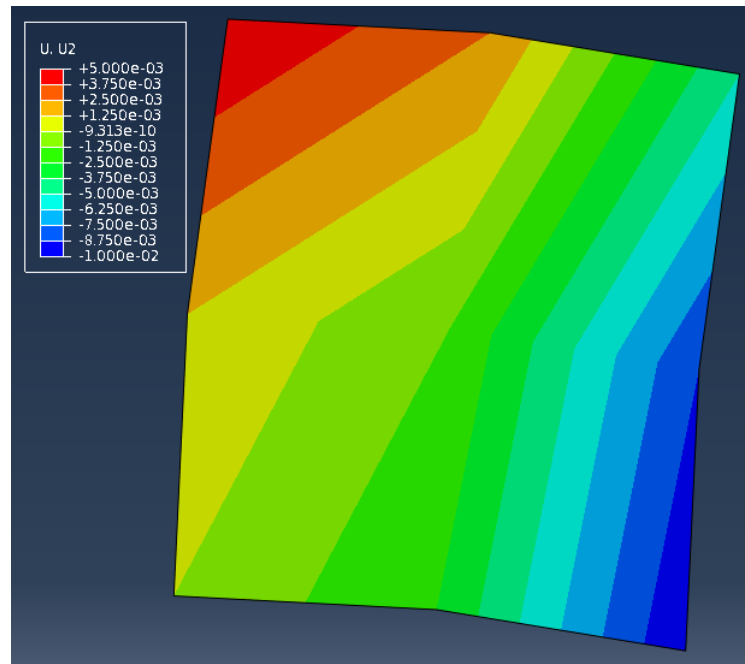

Figure 24: Visualization of Displacement u_2 in Patch Test-1 Model.

7.2   Patch Test - 2

Now, we considered the same geometry with same dimensions as described in Patch test-1, but now we mesh it into four element as shown in below fig.
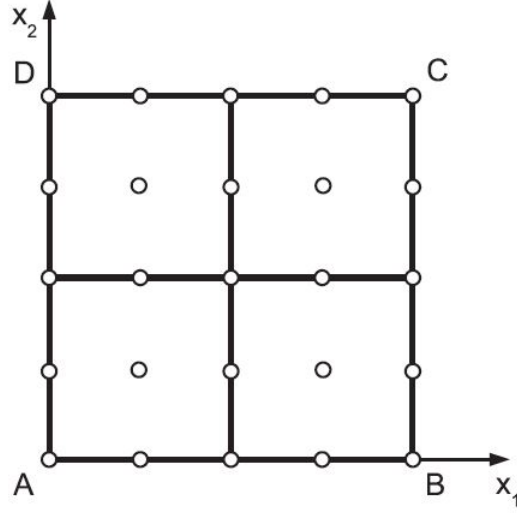
Figure 25: Square Domain with four element for implementation of Patch Test-2.

Now, we trying to implement the Patch test-2, for this one there are some criteria to pass the count test therefore, we considered the four element on the same geometry instead of one. In this patch test we just find the displacements and relaxed strain of the nodes at the boundary by analytical way as same as in patch test one.

In details, we have total 32 displacements values of all nodes and 32 relaxed strains values at each corner nodes of the elements which are prescribed in the Input file for four element in terms of boundary conditions.

As same as in Patch Test-1 i wrote the one python script for patch test-2 which also gives the analytical values of displacements and relaxed strains of the deformed nodes. we just give the values of it at the boundary and we compare the values of all inner nodes with analytical solutions.

Figure 26: Visualization of Displacement u_1 in Patch Test-2 Model.

Figure 27: Visualization of Displacement u_2 in Patch Test-2 Model.

A   Python file for Patch test-1

```python
### Patch Test for 1 element required some Analytical values of
### displacements and relaxed strain.
### So, here we use this equation to get the analytical values.

import numpy as np

### Initialize List for the new coordintes of the deformed nodes.
New_displacement = []
New_Relaxed_Strain = []

### Analytical Equations for displacements u1 and u2 of nodes :
def Displacement(x,y):
    u1 = -0.005*x*x + 0.01*y*y
    u2 = -0.01*x*x+0.005*y*y
    return [u1,u2]

### Analytical Equations for relaxed strain psi11,psi21,psi12 and psi22 of nodes :
def Relaxed_Strain(x,y):
    psi_1 = -0.01*x
    psi_2 =  0.02*y
    psi_3 = -0.02*x
    psi_4 =  0.01*y
    return [psi_1,psi_2,psi_3,psi_4]


### X and Y coordinates of the all nine nodes of the Element :
x_disp = np.array([0,1,1,0,0.5,1,0.5,0,0.5])
y_disp = np.array([0,0,1,1,0,0.5,1,0.5,0.5])

### X and Y coordinates of the corner nodes of the Element :
x_strain = np.array([0,1,1,0])
y_strain = np.array([0,0,1,1])

### Find the new coordintes of the deformed nodes :
New_displacement = Displacement(x_disp,y_disp)

### Find the relaxed strain values at corner nodes :
New_Relaxed_Strain = Relaxed_Strain(x_strain,y_strain)

### Print the Old_values of the nodes coordintes:
print("The old values of nodes coordintes x_old :")
print(x_disp)
print()
print("The old values of nodes coordintes y_old :")
print(y_disp)
print()

### Print the New_values of the nodes coordintes:
print("The New values of displaced nodes coordintes x_new and y_new :")
print(New_displacement)
print()

### Print the Relaxed strain values at the corner nodes.
print("Relaxed strain values at each corner nodes Psi_11,Psi_21,Psi_12,Psi_22 :")
print(New_Relaxed_Strain)
```
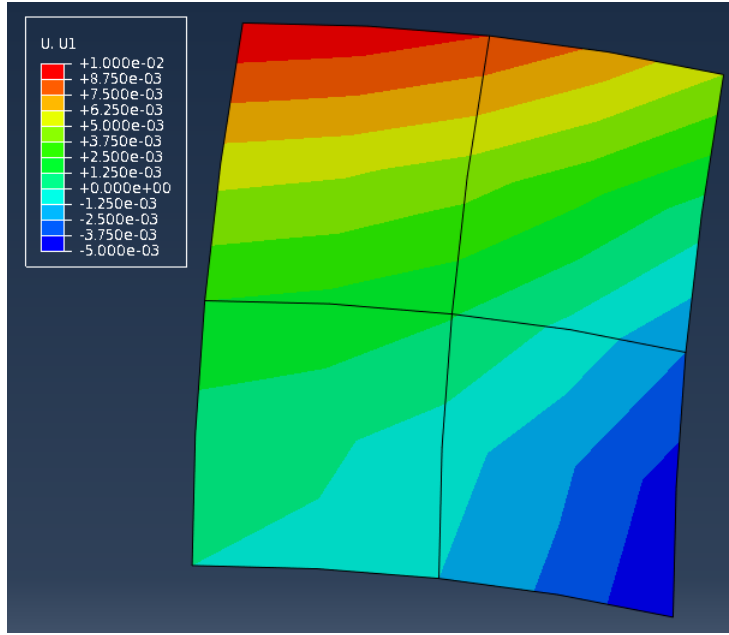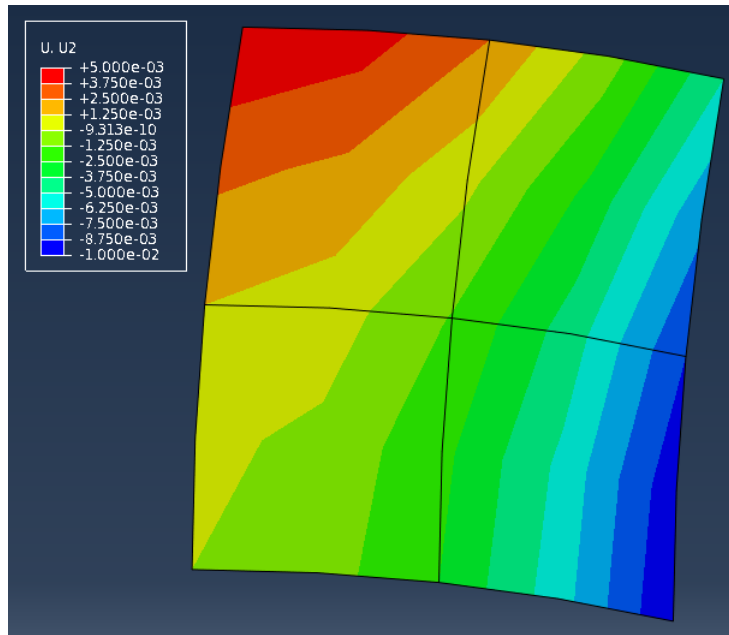
```python
import numpy as np
import matplotlib.pyplot as plt
import math as math
import csv
from copy import deepcopy
import csv


A = np.zeros([38,38])
RHS = np.zeros([38,1])
U = np.zeros([38,1])

################################################################################
# open file
with open('amtrx.dat') as file:

# loop over each line in file
    for i,line in enumerate(file.readlines()):

    # create a list of strings splitted by whitespace
        numbers = line.split()

        # loop over potential numbers
        for j,number in enumerate(numbers):
            number = float(number)
            A[i,j] = number

################################################################################
# open file
with open('rhs.dat') as file_1:

# loop over each line in file
    for i,line in enumerate(file_1.readlines()):

    # create a list of strings splitted by whitespace
        number = line.split()
        number = float(number[0])
        RHS[i,0] = number


################################################################################
# open file
with open('U.dat') as file_2:

# loop over each line in file
    for i,line in enumerate(file_2.readlines()):

    # create a list of strings splitted by whitespace
        number = line.split()
        number = float(number[0])
        U[i,0] = number

################################################################################
K_inv = np.linalg.inv(A)
Residual = np.dot(A,U) - RHS
print(np.dot(A,U))
```

B   Python file for Patch test-2

```python
### Patch Test for 4 element required some Analytical values of
### displacements and relaxed strain at the boundary and inner nodes.
### So, here we use this equation to get the analytical values.

import numpy as np

New_boundary = []
New_strain_boundary = []
New_inner = []
New_universe = []

### Analytical Equations for displacements u1 and u2 of nodes :
def Displacement(x,y):
    u1 = -0.005*x*x + 0.01*y*y
    u2 = -0.01*x*x+0.005*y*y
    return [u1,u2]

### Analytical Equations for relaxed strain psi11,psi21,psi12 and psi22 of nodes :
def Relaxed_Strain(x,y):
    psi_1 = -0.01*x
    psi_2 =  0.02*y
    psi_3 = -0.02*x
    psi_4 =  0.01*y
    return [psi_1,psi_2,psi_3,psi_4]

### X and Y coordinates of the all boundary nodes of the Element :
x_boundary = np.array([0,0.25,0.5,0.75,1,1,1,1,0.75,0.5,0.25,0,0,0,0])
y_boundary = np.array([0,0,0,0,0,0.25,0.5,0.75,1,1,1,1,1,0.75,0.5,0.25])

### X and Y cooedintes of all boundary corner nodes of the Element :
x_strain_boundary = np.array([0,0.5,1,1,1,0.5,0,0])
y_strain_boundary = np.array([0,0,0,0.5,1,1,1,0.5])

### X and Y coordinates of the all inner nodes nodes of the Element :
x_inner = np.array([0.25,0.5,0.75,0.25,0.5,0.75,0.25,0.5,0.75])
y_inner = np.array([0.25,0.25,0.25,0.5,0.5,0.5,0.75,0.75,0.75])

### X nad Y coordintes of the nodes which is occured in all elements :
x_universe = 0.5
y_universe = 0.5

### Find the new coordintes of the deformed nodes at boundary:
New_boundary = Displacement(x_boundary,y_boundary)

### Find the relaxed strain values at corner nodes at boundary :
New_strain_boundary = Relaxed_Strain(x_strain_boundary,y_strain_boundary)

### Find the new coordintes of the deformed nodes at inner side:
New_inner = Displacement(x_inner,y_inner)

### Find the relaxed strain values at corner nodes at inner side :
New_universe = Relaxed_Strain(x_universe,y_universe)

### Print the Old_values of the nodes coordintes at boundary:
print("The old values of boundary nodes coordintes x_old :")
print(x_boundary)
print()
print("The old values of boundary nodes coordintes y_old :")
print(y_boundary)
print()
print()

### Print the New_values of the displaced nodes coordintes at boundary:
print("The New values of displaced nodes coordintes x_new and y_new :")
print(New_boundary)
print()
```

```python
print()

### Print the relaxed strain values of the nodes at the boundary :
print("The values of displaced nodes relaxed strain psi11,psi21,psi12,psi22 :")
print(New_strain_boundary)
print()
print()

### Print the Old_values of the nodes coordintes at innerside :
print("The old values of innerside nodes coordintes x_old :")
print(x_inner)
print()
print("The old values of innerside nodes coordintes y_old :")
print(y_inner)
print()
print()

### Print the New_values of the displaced nodes coordintes at inner side:
print("The New values of displaced nodes coordintes x_new and y_new :")
print(New_inner)
print()
print()

### Print the relaxed strain values of the nodes at the middle :
print("The values of displaced nodes relaxed strain psi11,psi21,psi12,psi22 :")
print(New_universe)
```

C   Python file for 9-node Element mesh generation

```python
import numpy as np
import matplotlib.pyplot as plt
import math as math
import csv
from copy import deepcopy

# load the nodes and element data of 8-noded element.

node_file = 'nodes.dat'
element_file = 'elements.dat'
nodes = np.loadtxt(node_file,delimiter=",")
elements = np.loadtxt(element_file,delimiter=",")

# find the shape of nodes and elements matrix.

nodes_shape = np.shape(nodes)
elements_shape = np.shape(elements)
a1 = nodes_shape[0]
a2 = nodes_shape[1]
b1 = elements_shape[0]
b2 = elements_shape[1]

# initialize the shape function array and assign the values to the variables.

shape_function = np.zeros(8)
xi = 0
omega = 0


# Assign the values to the shape function array.

shape_function[0] = (1/4)*(1-xi)*(1-omega)*(-xi-omega-1)
shape_function[1] = (1/4)*(1+xi)*(1-omega)*(+xi-omega-1)
shape_function[2] = (1/4)*(1+xi)*(1+omega)*(+xi+omega-1)
shape_function[3] = (1/4)*(1-xi)*(1+omega)*(-xi+omega-1)
shape_function[4] = (1/2)*(1-xi**2)*(1-omega)
shape_function[5] = (1/2)*(1+xi)*(1-omega**2)
shape_function[6] = (1/2)*(1-xi**2)*(1+omega)
shape_function[7] = (1/2)*(1-xi)*(1-omega**2)

# initilize the new nodes and elements matrix.

New_nodes = np.zeros((a1+b1,a2))
New_elements = np.zeros((b1,b2+1))

# assign the same nodes matrix values to the new node matrix.

for i in range(a1):
    for j in range(a2):
        New_nodes[i,j] = nodes[i,j]

# assign the same element matrix values to the new element matrix.

for i in range(b1):
    for j in range(b2):
        New_elements[i,j] = elements[i,j]

# loop to assign the addition nodes to the new nodes matrix.
# loop to add the nodes number to the new element matrix.

for i in range(a1,a1+b1,1):
    New_nodes[i,:] = 0
    New_nodes[i,0] = i+1
    element_number = i - a1 + 1

# loop to find coordinate of the new ninth node of the element.
```

```python
    for j in range(1,8+1,1):
        node_number = elements[element_number-1,j]
        node_number = np.int(node_number)
        New_nodes[i,1] = New_nodes[i,1] + shape_function[j-1]*nodes[node_number-1,1]
        New_nodes[i,2] = New_nodes[i,2] + shape_function[j-1]*nodes[node_number-1,2]
    New_elements[element_number-1,9] = i+1

# convert new element matrix into integer matrix from folat matrix.

New_elements = New_elements.astype(int)
with open('New_nodes.dat', 'w') as file:
    for i in range(New_nodes.shape[0]):

# write the matrix column vice to convert the first column of new node matrix.
# into the integer since it contains the node number.

        file.write(str(int(New_nodes[i, 0])) + ',' + str(New_nodes[i, 1]) + ',' + str(New_nodes[i, 2]) + '\n')

with open('New_elements.dat', 'w', newline='') as file1:
    writer = csv.writer(file1)
    writer.writerows(New_elements)
```
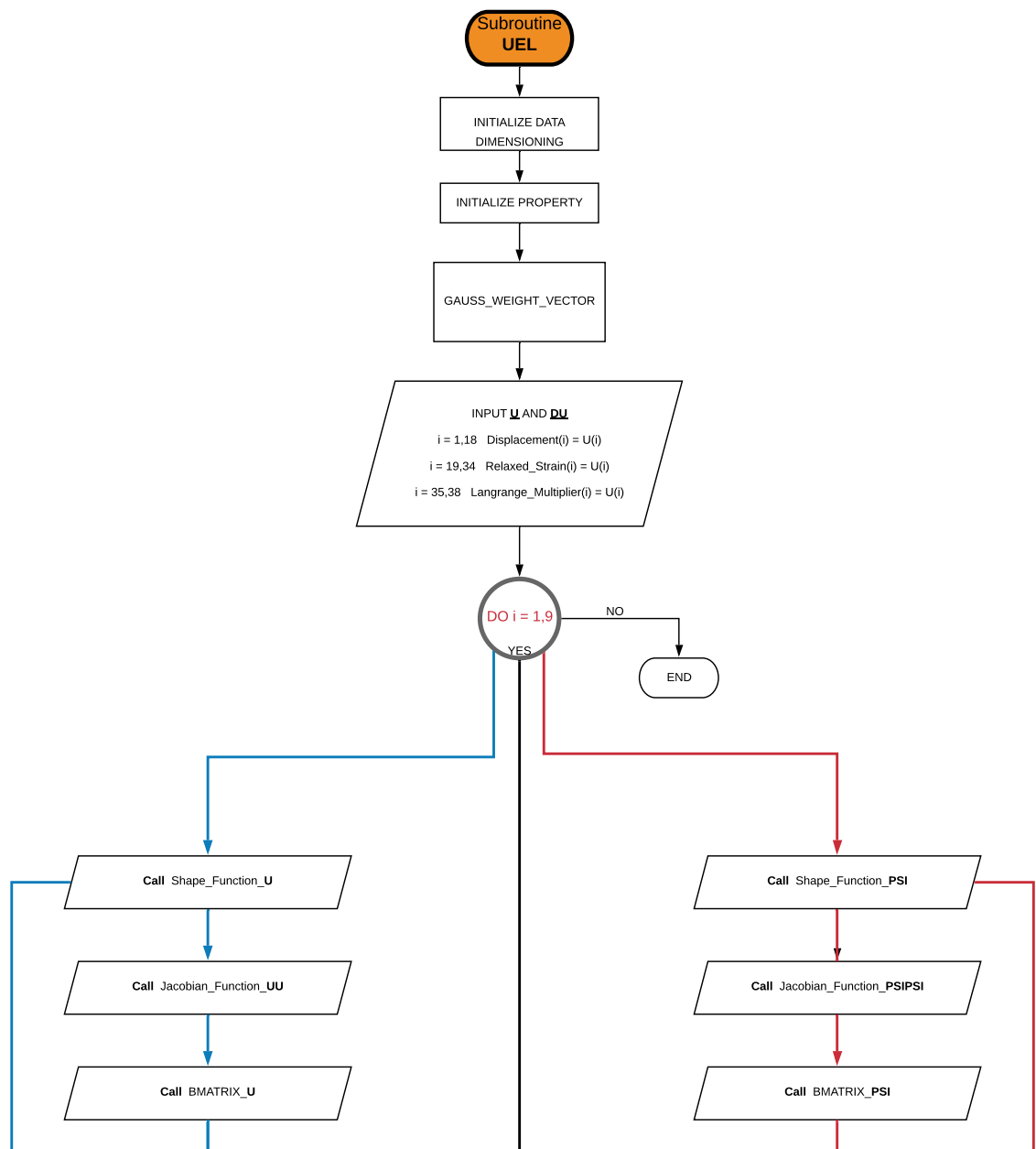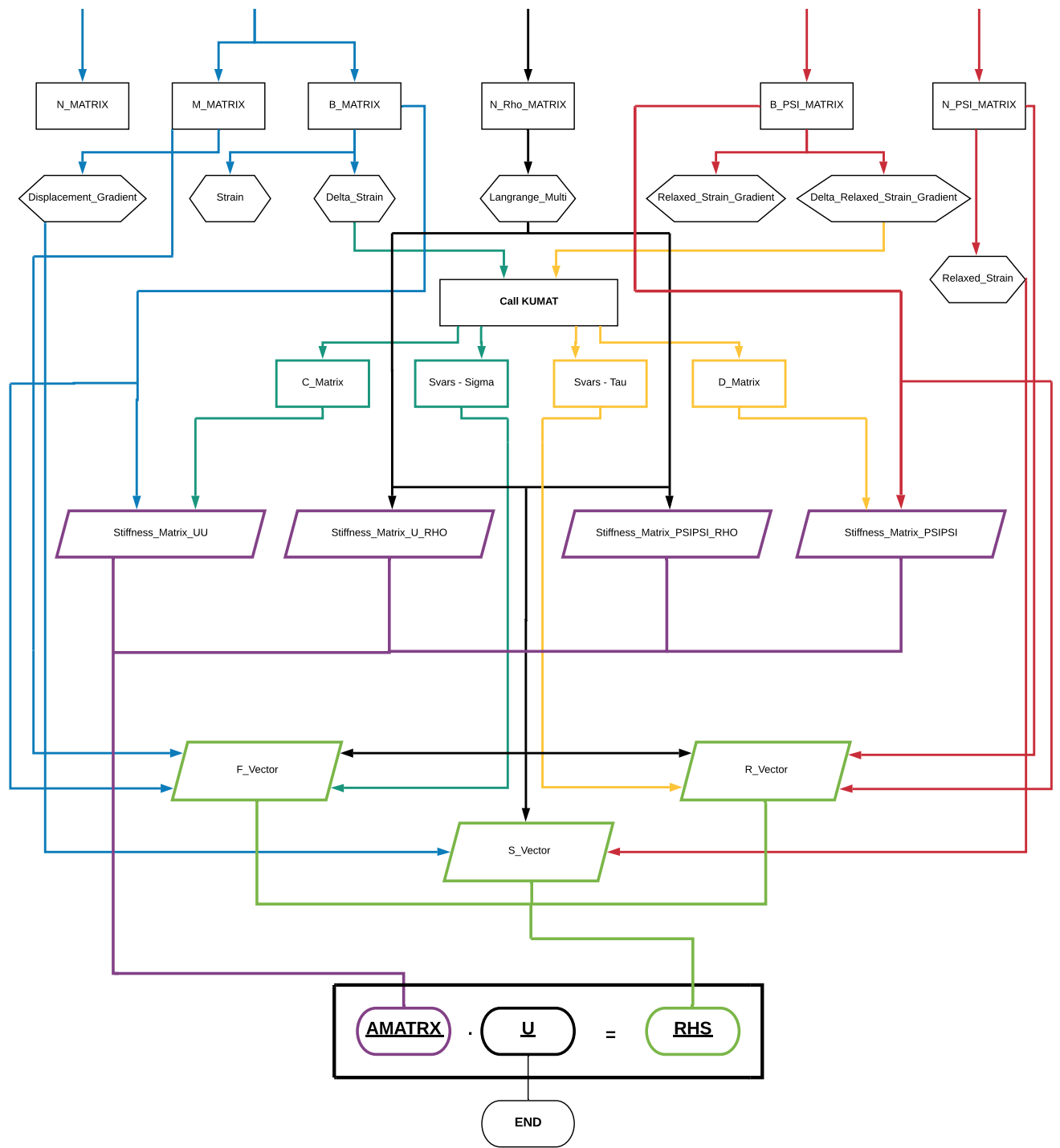
D   Flowchart of Strain Gradient Elasticity FEM Model

References

[1] E Amanatidou and N Aravas. Mixed finite element formulations of strain-gradient elasticity problems. *Computer Methods in Applied Mechanics and Engineering*, 191(15-16):1723–1751, 2002.

[2] Harm Askes and Elias C Aifantis. Gradient elasticity in statics and dynamics: an overview of formulations, length scale identification procedures, finite element implementations and new results. *International Journal of Solids and Structures*, 48(13):1962–1990, 2011.

[3] Tomislav Lesičar, Zdenko Tonković, and Jurica Sorić. Two-scale computational approach using strain gradient theory at microlevel. *International Journal of Mechanical Sciences*, 126:67–78, 2017.

[4] Raymond David Mindlin and NN Eshel. On first strain-gradient theories in linear elasticity. *International Journal of Solids and Structures*, 4(1):109–124, 1968.

[5] Filip Putar, Jurica Sorić, Tomislav Lesičar, and Zdenko Tonković. Damage modeling employing strain gradient continuum theory. *International Journal of Solids and Structures*, 120:171–185, 2017.

[6] John Y Shu, Wayne E King, and Norman A Fleck. Finite elements for materials with strain gradient effects. *International Journal for Numerical Methods in Engineering*, 44(3):373–391, 1999.