Numerical Study on Cohesive Zone Elements for Static and Time Dependent Damage

and its Application in Pipeline Failure Analysis

by

Vishal Chandrasekhar

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved July 2016 by the
Graduate Supervisory Committee:

Yongming Liu, Chair
Hanqing Jiang
Jay Oswald

ARIZONA STATE UNIVERSITY

August 2016

ABSTRACT

Cohesive zone model is one of the most widely used model for fracture analysis, but still remains open ended field for research. The earlier works using the cohesive zone model and Extended finite element analysis (XFEM) have been briefly introduced followed by an elaborate elucidation of the same concepts. Cohesive zone model in conjugation with XFEM is used for analysis in static condition in order to check its applicability in failure analysis. A real time setup of pipeline failure due to impingement is analyzed along with a detailed parametric study to understand the influence of the prominent design variable. After verifying its good applicability, a creep model is built for analysis where the cohesive zone model with XFEM is used for a time dependent creep loading. The challenge in this simulation was to achieve coupled behavior of cracks initiation and propagation along with creep loading. By using Design of Experiment, the results from numerical simulation were used to build an equation for life prediction for creep loading condition. The work was further extended to account for fatigue damage accumulation for high cycle fatigue loading in cohesive elements. A model was conceived to account for damage due to fatigue loading along within cohesive zone model for cohesive elements in ABAQUS simulation software. The model was verified by comparing numerical modelling of Double cantilever beam under high cycle fatigue loading and experiment results from literature. The model was also applied to a major industrial problem of blistering in Cured-In-Plane liner pipelines and a demonstration of its failure is shown. In conclusion, various models built on cohesive zone to address static and time dependent loading with real time scenarios and future scope of work in this field is discussed.

DEDICATION

To one who loves me with all her heart, mom

To one who sacrificed a lot to make me what I am, dad

To one who guides and advises me in life, brother

To one who teaches and imparts knowledge, teachers

ACKNOWLEDGMENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

One of the most imperative models for the analysis of fracture and damage of structural materials is the cohesive zone model. There has been numerous work in cohesive zone model due to good agreement with experimental results for various numerical studies. The main advantage of the cohesive zone model is that it can avoid the singularity during the numerical simulation of discontinuities and can include both crack initiation and propagation in a single framework. This work consists of two parts. The first part is using cohesive zone model (CZM) in conjugation with Extended Finite Element Method (XFEM) for a static load case to study pipeline cracking due to rock impingement, this is followed by a creep analysis using the same CZM with XFEM to predict the life to failure of pipe structure. The second part consists of incorporating a model accounting for fatigue damage due to high cycle loading in cohesive zone model. Then, the model was used to simulate Double Cantilever beam to verify its agreement with experimental results from the literature. The modified cohesive zone model to account for fatigue damage was applied to a real time problem in the multilayered pipeline.

Cohesive zone models have been used extensively in past decade in numerical simulations. Pablo [1] used cohesive model for shell elements to model crack propagation in thin walled structures wherein he showed simple 3D plate crack propagation and studied the variation due to shell and solid elements in analysis. Park et al. [2] explained the traction separation based cohesive zone model and the different relationship with respect to fracture surface. An XFEM based approach has been used for a long time by many researchers to obtain

information on crack propagation of the materials under study. Ebrahimi et al [3] used an XFEM to study crack analysis in composite media and determined the mixed mode stress intensity factor based on interactive integral approach to determine J integrals which were in good agreement to values of other models.

Many models have been developed using XFEM due to their advantages in handling discontinuity and cracks, further they do not require remeshing like other FEM analysis. Yazdani et al [4] gave modelling and analysis of delamination in composite laminates by combining the lower-order plate theory and the novel XFEM technique, and this model is able to accurately calculate the delamination onset and the propagation with less computational effort.

In Chapter 2, XFEM + CZM is used as an analysis methodology for static analysis to understand the working of the methodology and also to show how well the conjugation works on real time pipe system problems. This is followed by a time-dependent loading analysis where, maximum load is maintained for a particular time period similar to creep loading and the crack propagation in pipeline is studied. A detailed Design of Experiment (DoE) is conducted to obtain the model along with the crack growth vs time plot for the model. Bouvard et al [5] studied the fatigue and creep-fatigue crack growth in single crystal super alloys using cohesive zone model by traditional cyclic loading method to predict the crack growth. Kyungmok Kim [6] has worked on Creep–rupture model of aluminum alloys using Cohesive zone approach to analyze the long-term creep using a time jump strategy in cohesive zone law. Pouria Hajikarimi [7] et al used extended finite element method for

two dimensional creep analysis of a linear viscoelastic medium using a 4-node rectangular element and also studied the effect of enrichment function on the analysis.

In chapter 3, a model to address the fatigue damage due to high cycle loading is developed without actual loading and unloading. An envelope is developed to account the effect of high cycle loading instead of actual cyclic loading. This is then incorporated in Cohesive element in Abaqus using UEL and then verified by using DCB model. The results give a good match with existing values from literature. Then this model is used for analysis of a imperative problem in piping system and results are shown. Yangjian Xu [8] et al had worked on damage accumulations in mixed-mode fatigue crack growth using cyclic cohesive zone model along with XFEM, where a cyclic load was applied and results were obtained. This work though, is a similar approach to that of Hallett et al [9] who developed a model for fatigue degradation using cohesive zone model using an enveloped model instead of conventional cyclic model.

2. NUMERICAL SIMULATION USING COHESIVE ZONE MODEL FOR POLYMER MATERIAL

2.1 General Cohesive zone model

The cohesive zone modeling (CZM) is one of the most widely used fracture phenomenon based on the separation of surfaces involved in cracking. It is defined by cohesive tractions that exists in between material surfaces which is resisted by crack propagation. The main advantage of this method is that it has no stress singularity at crack tip. Unlike conventional methods, size of non-linear zone around crack tip does not have to be negligible and further it can also predict behavior of un-cracked structures.

The cohesive zone model has different governing equations but the most predominant one is traction-separation response model. This is a bilinear curve between the traction force and separation. This curve has two main parts; a damage initiation criterion and a damage evolution law. The initial response is assumed to be linear till damage initiation. However, once a damage initiation criterion is met, damage can occur according to a user-defined damage evolution law.
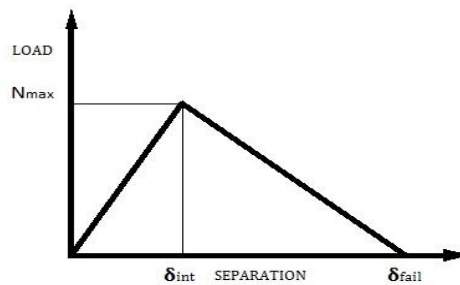
Figure 1: Traction separation response curve

Damage initiation is the first linear part of the bilinear curve which is governed by maximum strength of the material, beyond which the degradation of the cohesive response begins. The process of degradation starts once the contact stresses and/or contact separations is beyond the damage initiation criterion. The damage initiation criterion can be set in terms of maximum strain and stress of the material. Each damage initiation criterion also has an output variable which indicates the criterion status, if the value is beyond 1 it means the damage initiation criterion has started. Damage initiation criteria not associated with evolution law can still be used to evaluate the propensity of the material to undergo damage without actually modeling the damage process (i.e., without actually specifying damage evolution).

The four ways to set the damage initiation criterion are listed as Eq (1) to (4): maximum stress, maximum separation, quadratic stress and quadratic separation. Damage is assumed to initiate when the maximum contact stress ratio reaches a value of one.

$$\max\left\{\frac{t_n}{t_n^o}, \frac{t_s}{t_s^o}, \frac{t_t}{t_t^o}\right\} = 1 \tag{1}$$

$$\max\left\{\frac{\delta_n}{\delta_n^o}, \frac{\delta_s}{\delta_s^o}, \frac{\delta_t}{\delta_t^o}\right\} = 1 \tag{2}$$

$$\left\{\frac{t_n}{t_n^o}\right\}^2 + \left\{\frac{t_s}{t_s^o}\right\}^2 + \left\{\frac{t_t}{t_t^o}\right\}^2 = 1 \tag{3}$$

$$\left\{\frac{\delta_n}{\delta_n^o}\right\}^2 + \left\{\frac{\delta_s}{\delta_s^o}\right\}^2 + \left\{\frac{\delta_t}{\delta_t^o}\right\}^2 = 1 \tag{4}$$

5

where t_n^o, t_s^o and t_t^o represent the maximum contact stress when the separation is either purely normal to the interface or purely in the first or the second shear direction respectively. Likewise, δ_n^o, δ_s^o and δ_s^o represent the maximum contact separation, when the separation is either purely along the contact normal or purely in the first or the second shear direction respectively.

The next part of traction separation is damage evolution law, which describes the rate at which the cohesive stiffness is degraded once the corresponding initiation criterion is reached. The general framework for describing the evolution of damage takes place in two stages: softening of the yield stress and degradation of the elasticity. A scalar damage variable D, represents the overall damage at the contact point. It initially has a value of 0. If damage evolution is modeled, D monotonically evolves from 0 to 1 upon further loading after the initiation of damage. The contact stress components are affected by the damage according to Eq (5) to (8)

$$t_n = \begin{cases} (1-D)\bar{t}_n & if \ t_n \geq 0 \\ \bar{t}_n & otherwise \end{cases} \tag{5}$$

$$t_s = (1-D)\bar{t}_s \tag{6}$$

$$t_s = (1-D)\bar{t}_s \tag{7}$$

There are mainly two components to define damage evolution: the first component is either specifying the effective separation at complete failure δ_f relative to the effective separation at the initiation of damage δ^o; or the fracture energy dissipated due to failure G^c. The

second component to the definition of damage evolution is the specification of the nature of the evolution of the damage variable D, between initiation of damage and final failure. This can be defined by linear or exponential softening laws or specifying D directly as a tabular function of the effective separation relative to the effective separation at damage initiation.

The Figure 2 shows the schematic representation of the dependence of damage initiation and evolution on the mode mix for a traction-separation response with isotropic shear behavior. The figure shows the traction on the X axis and the magnitudes of the normal and the shear separations along the two Y axes. The unshaded triangles in the two vertical coordinate planes represent the response under pure normal and pure shear separation, respectively. All intermediate vertical planes (that contain the vertical axis) represent the damage response under mixed-mode conditions with different mode mixes. The dependence of the damage evolution data on the mode mix can be defined either in tabular form or, in the case of an energy-based definition, analytically.
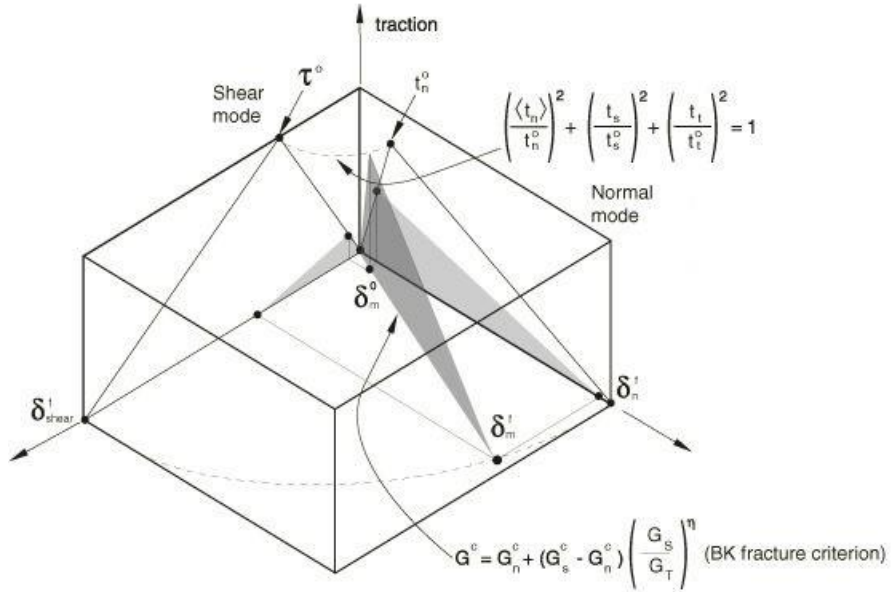
Figure 2[24]: Schematic Diagram showing mixed mode traction separation response

The cohesive zone model can be implemented by either using cohesive elements or cohesive surfaces. Though they follow the same bilinear curve there are some differences. The cohesive surfaces use the material properties as interaction properties where initial stiffness is contacted automatically. In cohesive element, the cohesive zone model is defined as material property along with special cohesive element definition. The initial stiffness has to be defined as a material property and is easier to manipulate according to requirement.

2.2 Extended Finite Element Analysis (XFEM)

The Finite Element Method (FEM) has been used for decades to analyze and solve imperative problems. There are number of instances where the usual FEM methodology have certain restrictions in efficiently solving some problems involving interior boundaries, discontinuities or singularities, because of the need of remeshing and high mesh densities. The Extended Finite Element Analysis (XFEM) is a latest development in the field of Finite Element Analysis in which a special enriched function is introduced in order to effectively model discontinuities, such as cracks, without conforming the mesh to the discontinued geometry. Only solid (continuum) elements can be associated with the enriched feature. XFEM is a methodology where local elements are partition of unity in order to account for the discontinuity.

$$\boldsymbol{u}^h(\boldsymbol{x}) = \sum_{n_I \in S} \phi_1(\boldsymbol{x})u_1 + \sum_{n_J \in S_\Psi} \phi_J(\boldsymbol{x})\,\Psi(\boldsymbol{x})\boldsymbol{a}_J \qquad (8)$$

The Extended Finite Element Analysis (XFEM) is defined by Eq (8) of which the first part is from FEM and second part is Enrichment function. XFEM extends the piecewise polynomial function space of conventional finite element methods with extra functions. The enrichment function can be divided into two parts, Heaviside enrichment term that accounts for jump discontinuity and crack tip enrichment function which handles the crack tip. The Eq (9) shows the enrichment function divided into Heaviside enrichment and crack tip enrichment.

$$\boldsymbol{u}^h(\boldsymbol{x}) = \sum_{n_I \in S} \phi_1(\boldsymbol{x})u_1 + \sum_{I \in N_\Gamma} \phi_1(\boldsymbol{x})H(\boldsymbol{x})\,\boldsymbol{a}_I + \sum_{I \in N_\Lambda} \phi_1(\boldsymbol{x})F_\alpha(\boldsymbol{x})b_I^\alpha \qquad (9)$$

$$H(x) = \begin{cases} 1, & if\ (x - x^*)\,n \geq 0 \\ -1, & otherwise \end{cases} \tag{10}$$

$$[F_\alpha(x), \alpha = 1 - 4] = \left[\sqrt{r}\sin\frac{\theta}{2}, \sqrt{r}\cos\frac{\theta}{2}, \sqrt{r}\sin\theta\sin\frac{\theta}{2}, \sqrt{r}\sin\theta\cos\frac{\theta}{2}\right] \tag{11}$$

In Eq (10) which represents the Heaviside enrichment function, x is an integration point, x* is the closest point to x on the crack face and n is the unit normal at x* and Eq (11) represents the crack tip enrichment function where (r, θ) denote coordinate values from a polar coordinate system located at the crack tip.

The crack tip and Heaviside enrichment functions are multiplied by the conventional shape functions thus the enrichment is locally associated around the crack. The crack location is defined by using level set method, where two function ϕ and Ψ are used to completely describe the crack. The function ϕ represents the crack face and interaction of ϕ and Ψ function denotes the crack front. These functions are defined by nodal values whose spatial variation is determined by using finite element shape functions. The function's values are specified only at nodes belonging to elements being cut by crack.

When the crack propagates through the element it cracks the element nodes and creates phantom nodes on both the divided elements. The discontinuous element with Heaviside enrichment is treated as a superposition of two continuous elements with phantom nodes. The Figure 3 below shows the creation of phantom nodes for a clear representation.
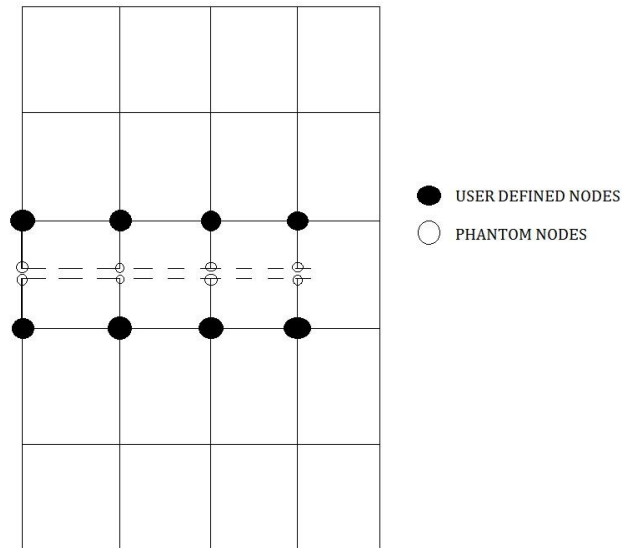
Figure 3: Phantom nodes creation and crack propagation

In Classical finite element method (FEM) adaptive meshing can be used for crack analysis, but it is very time consuming for large system analysis. Since objective is to develop an efficient and accurate failure analysis and prediction, XFEM in conjugation with Cohesive zone model is used. In FEM there are many methods to simulate crack propagation but due to high accuracy requirement and ease of computation we choose XFEM over FEM. The main reason to use XFEM + CZM is that the use of CZM will enable predictions under complex loading conditions and for different failure modes. XFEM allows non-matching mesh with respect to the crack and allows the arbitrary tracking of the CZM within the simulation domain.

11

2.3 Numerical simulation of Surface Cohesive Zone Model for static loading

The numerical simulation to study cohesive zone model has been done using ABAQUS software as it gave the freedom to use both XFEM and CZM technique. The most important parameters to define the model in ABAQUS are pipe dimensions, material model, loading and boundary condition. In the following section, modeling technique used for pipeline model, material model used for XFEM + CZM, loading and boundary condition to simulate a real time problem has been elaborated in detail.

2.3.1 Modeling Technique

In this part, the numerical modeling was done using ABAQUS for a pipe model with close to real time dimensions. As XFEM + CZM simulation takes longer to analyze a full model analysis leading to an increased computation cost and time, only a quarter of the model is modelled along with the region of interest like the pipe impingement. The results for a quarter model will be same as full scale model but the modeling uses lesser number of elements along with appropriate boundary conditions. The model was meshed using various control parameters such as imprinting faces, edge biasing, edge divisions etc. to achieve good quality elements in the desired region. Tet elements were used as model had a curved surface. The meshing is done more near the region of interest so as to get a more accurate result. After meshing, material model was assigned which is described in the next section. The modeling is done in two stages were the pipe is first modelled with a

dimension of outer diameter of 120 mm and inner diameter of 100 mm and an impingement of 20 mm sphere which is divided into half along YZ axis and XZ axis respectively.
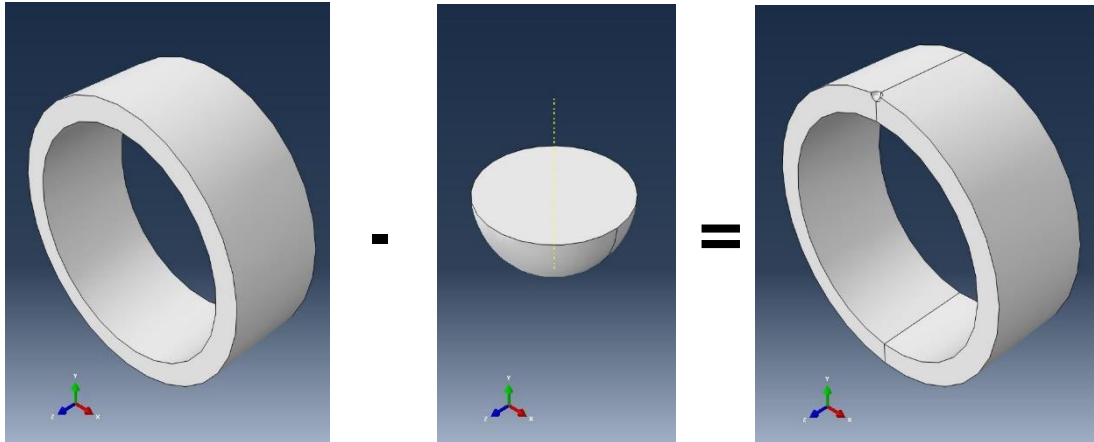
Figure 4: Modelling of pipeline structure

The circular section in Figure 4 represents the pipeline and the half sphere represents the impingement and are created as separate parts and then assembled to form the model.

2.3.2 Material Model

The material model has three main parts: property, element type and section assignment. The section assignment of each part is solid section and the element type is 3D stress element. The material used is PA11 and the values are obtained from literature [10]. The properties associated with the model is listed below in the table

Table 1: Properties of material PA11

S.No	Property of Material	Value
1	Modulus of Elasticity	1700 MPa
2	Poisson's Ratio	0.4
3	Utimate Tensile Strength	68.9 MPa

4	Yield Tensile Strength	44 MPa
5	Max. Principal Stress value for CZM	62 MPa
6	Fracture Energy for CZM	620 KJ/m^2

2.3.3 Loading and boundary condition

The loading and boundary condition are imperative part to any analysis, they are key to an accurate analysis. The boundary conditions are set such that they are close to real time applications so as to narrow the difference between the actual occurrence and simulation result. There are some universal rules for giving boundary conditions.

- BCs can only be applied in directions that the element has DoF's.

- Every DoF must be restrained in at least one place in the model.

- There must be at least one applied load to get a solution.

The internal pressure of the pipeline was maintained as 30 MPa and load due to rock impingement was 15 MPa. The figure below shows the boundary condition given to the model.
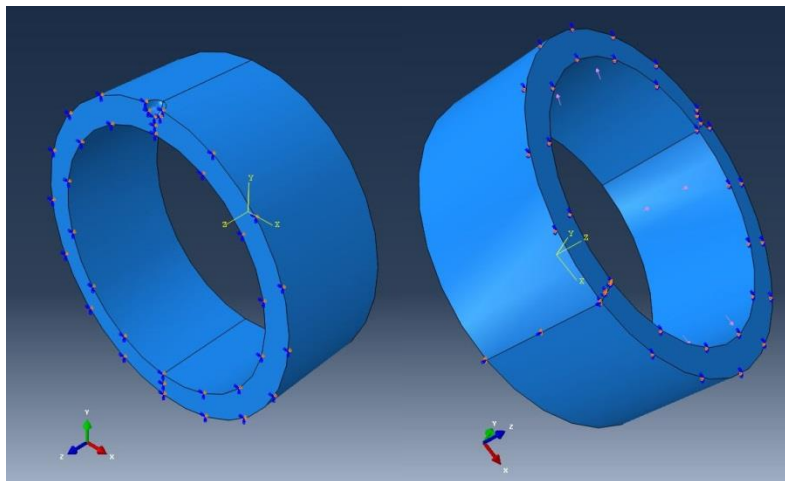
Figure 5: Boundary condition given to model

2.3.4 Simulation results for static analysis

The simulation results can be viewed in two perspectives: crack propagation which can be obtained by Status of XFEM that shows visual of crack growth along with the indication of elements failure, Von - mises stress which gives an idea about the maximum stress along the region of interest. This is an analysis for visualization of the failure giving us the ultimate load that the material can bear before failure. Figure 7 and 8 shows stages of crack propagation and von - mises stress respectively.
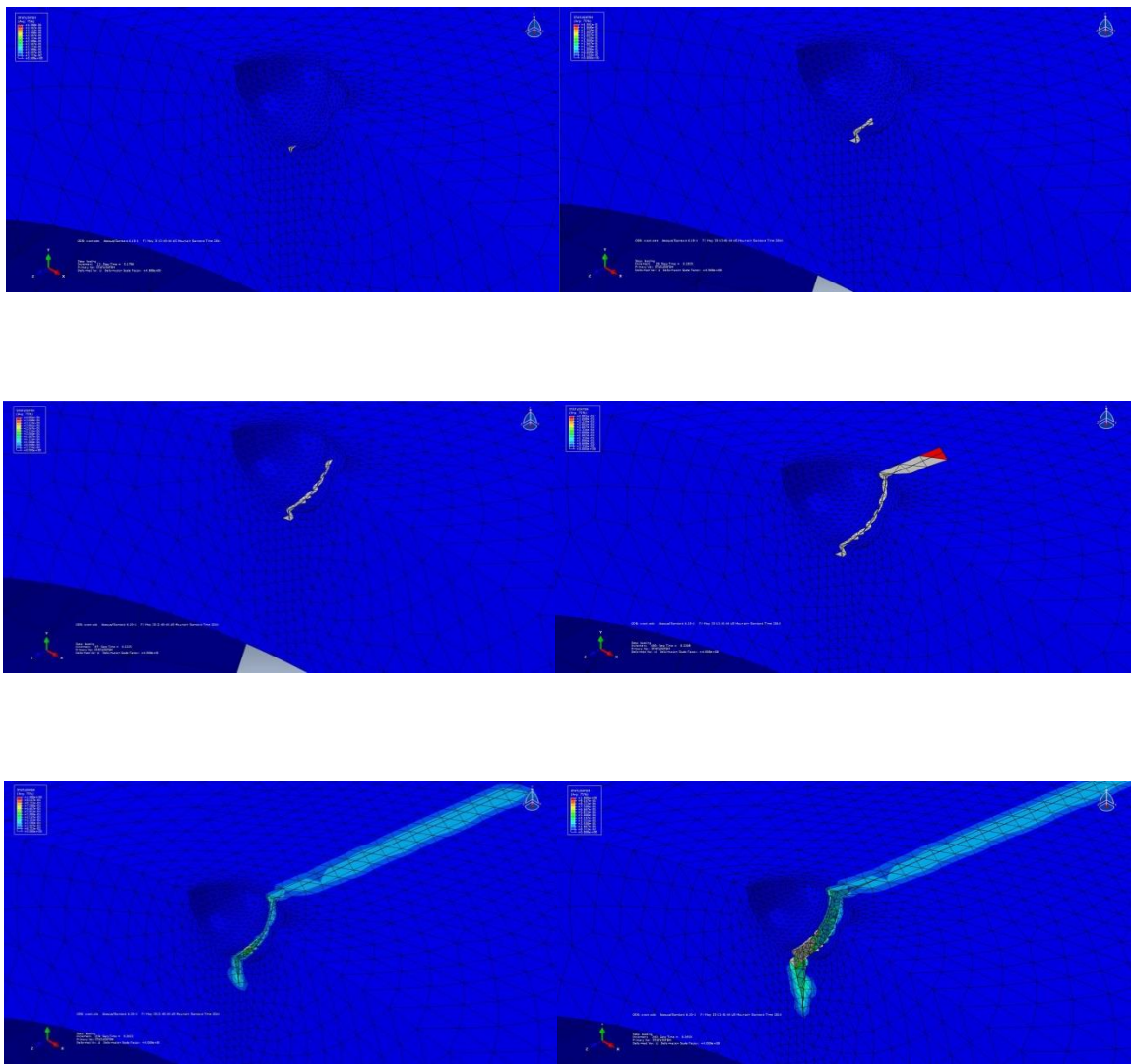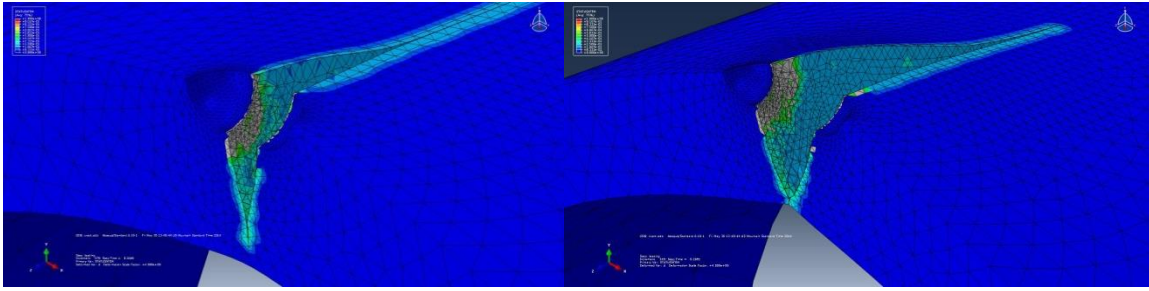
Figure 6: Crack Propagation
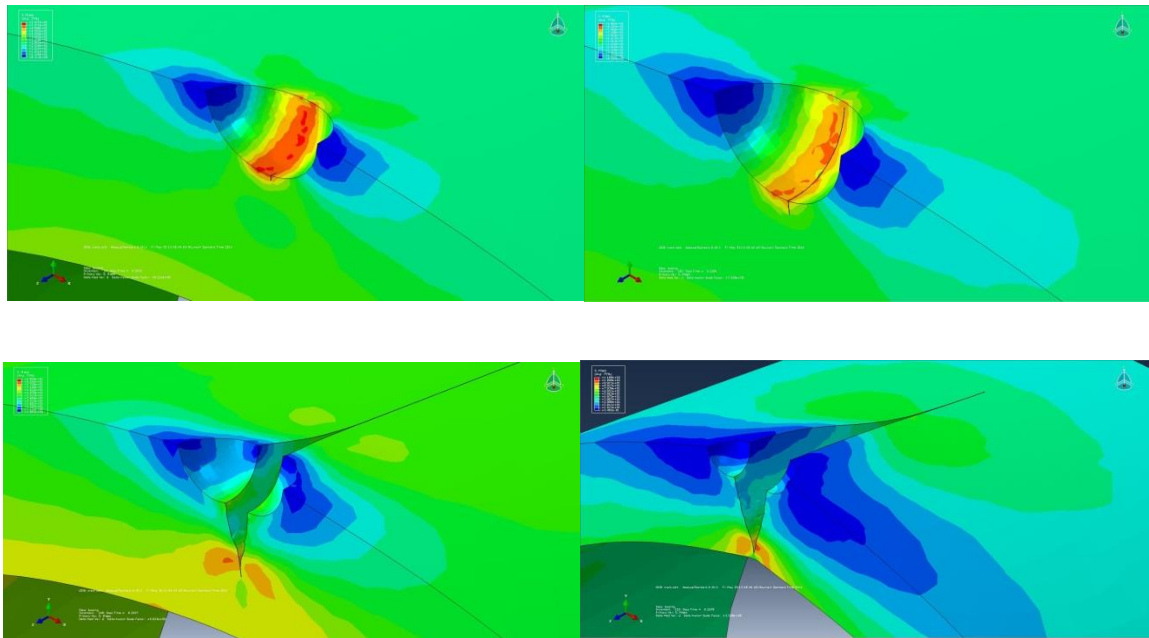
Figure 7: Von – mises Stress Plots

2.3.5 Parametric Study on Static analysis

The static analysis gives us an idea about all the parameters of the analysis and for the next time dependent analysis it is important to determine the important parameters so as to close the difference between actual value and analysis values. Thus a detailed parametric studies is carried out by considering three major parameters: Internal Diameter, Pipe Thickness, Cavity (impingement diameter).

These parameters were varied individually while keeping others constant to understand the impact of that particular parameter under analysis. This technique is called blocking. It is suggested that only trends of the graphs should be observed and not the values since the values chosen for material model were taken from literature available and not from any material manufacturer. All the graphs are plotted for Pi and Pc where Pi indicates the pressure at which the crack first appears in the model. Pc stands for critical pressure at which crack grows throughout the thickness. Deformation mentioned in the graphs is the deformation at a point on the inner surface of the pipe indicating the amount by which pipe has expanded.

2.3.5.1 Internal Diameter

The internal diameter was varied over a range to understand its impact on the crack initiation and propagation. Attached below is the data compiled from the simulations.

Table 2: Parametric study - Internal Diameter

Parameter	Graph1			Graph2		
	P1		U1	P2		Para value
Internal Diam.	0.6718	8.0616		0.8675	10.41	60
	0.4918	5.9016	2.07792	0.6118	7.3416	80
	0.3818	4.5816	2.34779	0.497	5.964	100
	0.3218	3.8616	2.97229	0.4255	5.106	120
	0.2818	3.3816	3.29145	0.3611	4.3332	140
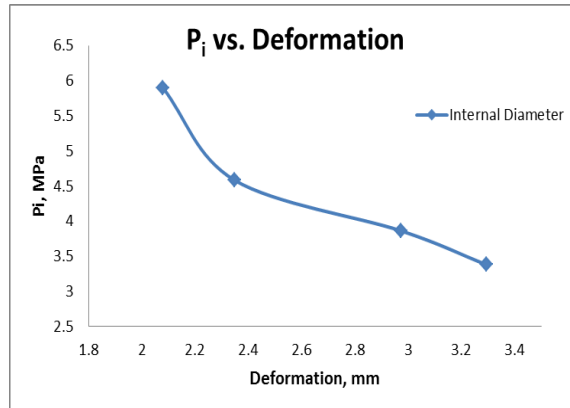
Figure 8: Pi vs. Deformation (Internal Dia.)
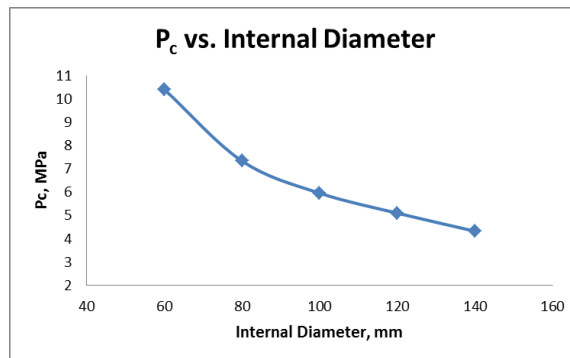
Figure 9: Pc vs. Internal Diameter

Both Pi and Pc decrease with parameter value indicating a negative effect of the parameter on the crack. Higher the value of internal diameter, higher is the risk of crack initiation and propagation.

2.3.5.2 Pipe Thickness

Pipe thickness was varied keeping internal diameter and loading constant. Attached below is the data compiled from the simulations.

Table 3: Parametric study – Pipe Thickness

Parameter	Graph1			Graph2		
	P1		U1	P2		Para value
	0.2818	3.3816	2.11463	0.3699	4.4388	15
	0.3818	4.5816	2.34779	0.497	5.964	20
Thickness	0.4818	5.7816	2.69947	0.6468	7.7616	25
	0.6118	7.3416	3.02949	0.772	9.264	30
	0.4318	8.636	3.19506			35
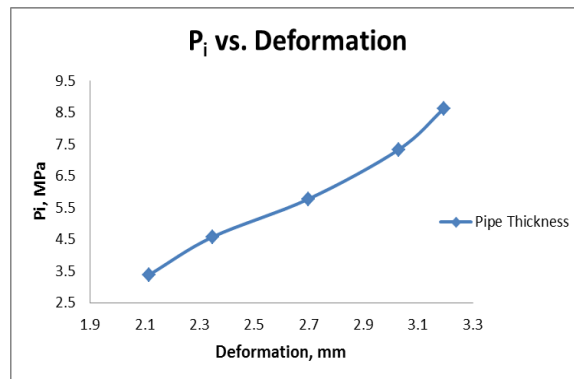
Figure 10: Pi vs. Deformation (Pipe thickness)
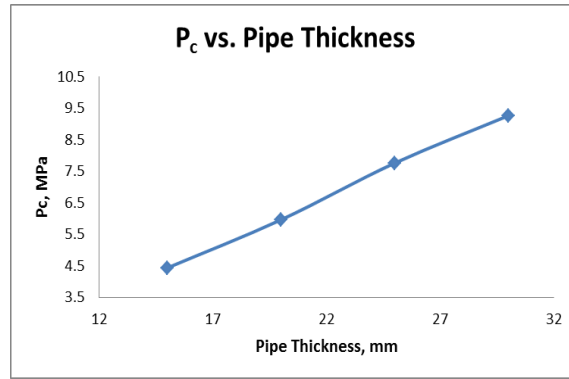
Figure 11: Pc vs. Pipe Thickness

From the graphs it can be observed that pipe thickness has a positive impact on crack. Thicker the pipe, higher will be the value of Pi and Pc.

2.3.5.3 Cavity Radius

Cavity radius was varied keeping internal and outer diameter and loading constant. Attached below is the data compiled from the simulations.

Table 4: Parametric study – Cavity Radius

Parameter	Graph1			Graph2		
	P1		U1	P2		Para value
Cavity Radius	0.3918	4.7016	2.36403			3.5
	0.3818	4.5816	2.34779	0.497	5.964	5
	0.3718	4.4616	2.33002	0.48	5.76	6.5
	0.3518	4.2216	2.31056	0.4362	5.2344	8
	0.3218	3.8616	2.30308	0.4218	5.0616	9.5
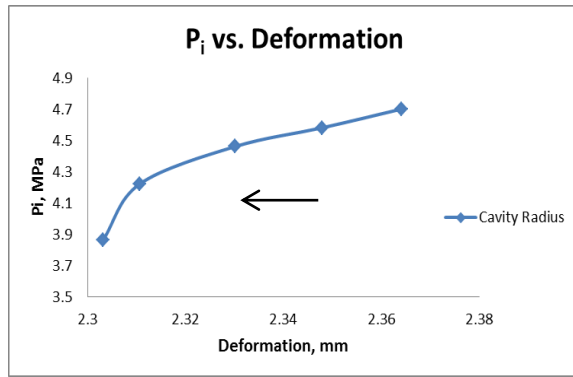
Figure 12: Pi vs. Deformation (Cavity Radius)
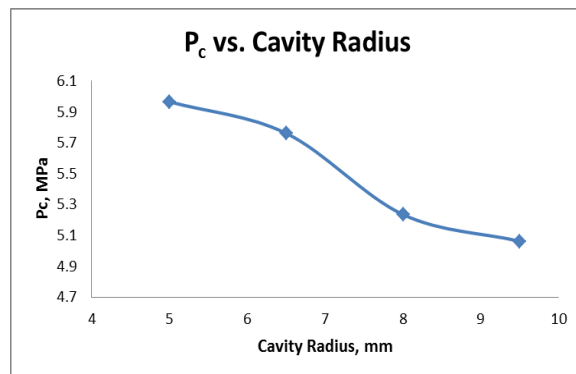
Figure 13: Pc vs. Cavity Radius

It can be observed from the graphs that cavity radius has a negative impact on the crack. The arrow marked in figure represents the direction of graph where data for first value of the parameter is represented by top right corner. The graph flows from top right corner to bottom left corner. Higher the value of cavity radius, higher will be the risk of crack.

2.4   Time dependent Creep Loading

Mechanical properties of polymers vary enormously with time and temperature. The time dependence complicates measurement of these properties since specific conditions under which experiments are carried out must necessarily be incorporated into the definition of appropriate parameters. Thus, the terms creep, stress relaxation and dynamic mechanical measurements are all common in this field [11]. The mechanical properties of a polymeric component are dominated by its viscoelasticity. This is reflected by the time-dependency of the mechanical response of a component during loading. Hence, a polymer behaves differently if subjected to short term or long term loads [12]. A brief review for the fundamental creep mechanics is briefly mentioned below.

A polymer, at a specific temperature and molecular weight, may behave as a liquid or a solid, depending on the speed (time scale) at which its molecules deform. This behavior, which ranges between liquid and solid, is generally referred to as the viscoelastic behavior or material response. Polymers are viscoelastic materials and exhibit time-dependent relaxations when subjected to stress or strain. While creep is a measure of increase in strain with time under a constant stress, stress relaxation is the reduction of stress with time under a constant strain. Therefore, creep and stress relaxation tests measure the dimensional stability of a polymer over time. Such tests are of great importance for investigated materials, particularly if a polymer must be in service under stress and strain for long periods [13]. The model for simulation is limited to linear viscoelasticity, which is valid for polymer systems undergoing small or slow deformations. Non-linear viscoelasticity is

required when modeling large rapid deformations, such as those encountered in flowing polymer melts. In linear viscoelasticity, the stress relaxation test is often used, along with the time temperature superposition principle and the Boltzmann superposition principle, to explain the behavior of polymeric materials during deformation [12].

When a plastic material is subjected to a constant load, it deforms continuously. The initial strain is roughly predicted by its stress-strain modulus. The material will continue to deform slowly with time indefinitely or until rupture or yielding causes failure. The primary region is the early stage of loading when the creep rate decreases rapidly with time. Then it reaches a steady state which is called the secondary creep stage followed by a rapid increase (tertiary stage) and fracture.
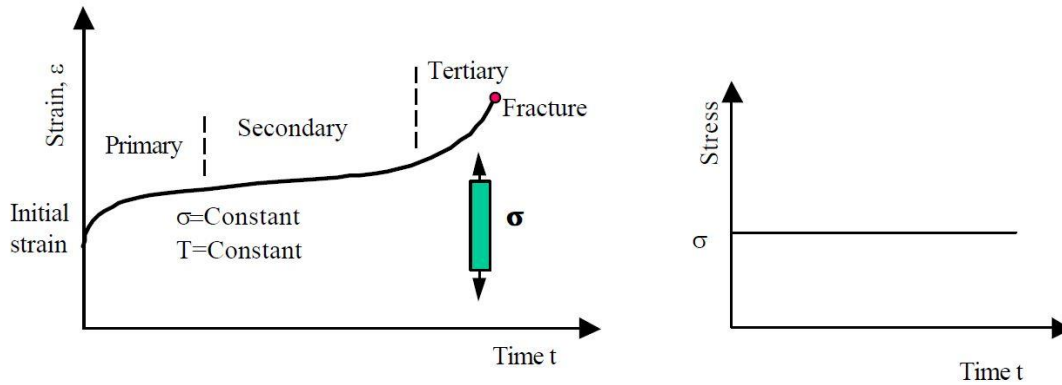
Figure 14[11]: Schematic illustration of creep behavior

If the applied load is released before the creep rupture occurs, an immediate elastic recovery equal to the elastic deformation, followed by a period of slow recovery is observed which is shown in Figure 15. The material in most cases does not recover to the

original shape and a permanent deformation remains. The magnitude of the permanent deformation depends on length of time, amount of stress applied, and temperature [14].
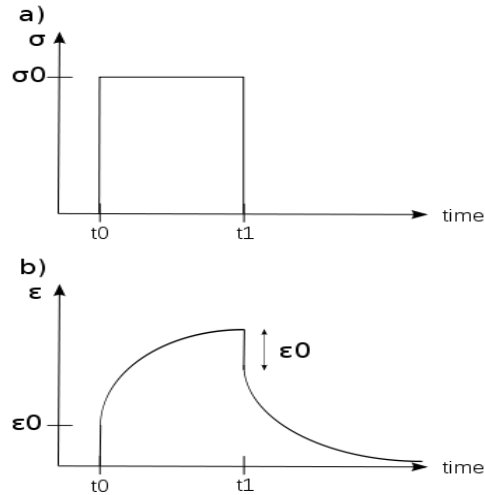
Figure 15[14]: Schematic illustration of strain recovery

In this numerical study, the same methodology of XFEM and cohesive zone which has been used for previous analysis is used along with certain modifications to the loading to get the time-dependent creep loading. For this the time period of the analysis was increased to a particular value say 10 which represents actual time period. The load is then applied such that the entire load is ramped up for the first time period and then maintained at this time period for rest of the complete time period. Figure 16 shows a representation of the time period vs load for a clearer illustration.
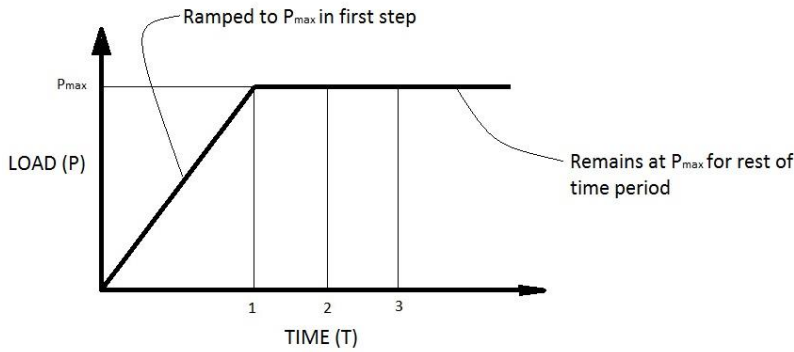
Figure 16: Load vs time plot for simulation model

2.5 Numerical study of cohesive zone model on creep loading

2.5.1 Model for Numerical analyses of creep loading

In the previous section, static fracture of Poly-Amide 11 was done and the focus was on

the failure strength prediction with existing impingement in representative pipe structure.

Also, thorough parametric studies were carried out on the design variables to understand it

pivotal role in failure of pipelines. The design variables were varied one at a time while

keeping others constant. This helped in screening the effect of each variable on the

response. The pressure at which crack initiates in the pipe was selected to be the response

variable. The failure mode was simulated using XFEM method and Cohesive Zone Model.

Moving forward a methodology is defined on how to simulate Slow Crack Growth under

creep behavior with given loading conditions. This investigation focuses on the time

dependent failure of pipeline structures. As done earlier in all simulations, XFEM and

cohesive zone model was also used. The challenge in this simulation was to achieve

coupled behavior of cracks initiation and propagation along with creep loading. This methodology is explained below and later the simulation result for pipe model is described.

The stress relaxation and the creep test are well-known long-term tests. The stress relaxation test is difficult to perform and is, therefore, often approximated by data acquired through the more commonly used creep test. The stress relaxation of a polymer is often considered the inverse of creep. The creep test, which can be performed either in shear, compression or tension, measures the flow of a polymer component under a constant load. It is a common test that measures the strain ε, as a function of stress, time and temperature. Standard creep tests such as DIN 53 444 and ASTM D2990 can be used. Creep tests are performed at a constant temperature, using a range of applied stress. On plotting creep data in a log-log graph, in the majority of the cases, the creep curves reduce to straight lines. Hence, the creep behavior of most polymers can be approximated with a power law where strain over time is based on stress 'σ' and time 't' and creep constant k(T) which depends on Temperature (T) [12,14]:

$$\varepsilon\,(t\,) = k(T)\,\sigma^m\,t^n \tag{12}$$

In this methodology, Creep properties are used for analysis where the Creep constant k(T) is given for a particular temperature and the loading is given for a particular time period. The full magnitude is applied within the first time period and then maintained for rest of time period in analysis.

2.5.2 Modelling Technique

The above discussion only serves as a general background for the simulation. The values used in numerical simulation are from literature. The full pipe model is developed but the properties of XFEM + CZM is applied to region of interest around the impingement region so as to save computational time and cost. The model is meshed by using Tet element due to curved shape of pipeline and are concentrated at the region of impingement so as to study the initiation and propagation of crack. After meshing the material model is applied with properties from literature for PA11 including creep properties which are listed in material model. Unlike the previous model, there is no cut due to impingement as the simulation is time-dependent and the load applied has to be for a long time period. Hence the region of impingement is taken as circular region of interest. The modeling is a pipe section with dimension of outer diameter of 120 mm and inner diameter of 100 mm and a circular section of diameter 20 mm on the top surface represents the area of impact of impingement, where maximum load is going to occur. A region of interest is created by segmenting the pipe section around the impingement circle by dividing 10 mm on both side of circle with XZ datum plane. The Figure 17 shows the modelled section used for numerical simulation for creep analysis.
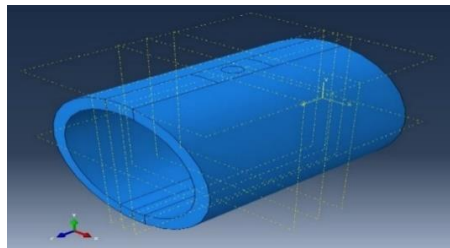
Figure 17: Pipe section for numerical simulation for creep loading
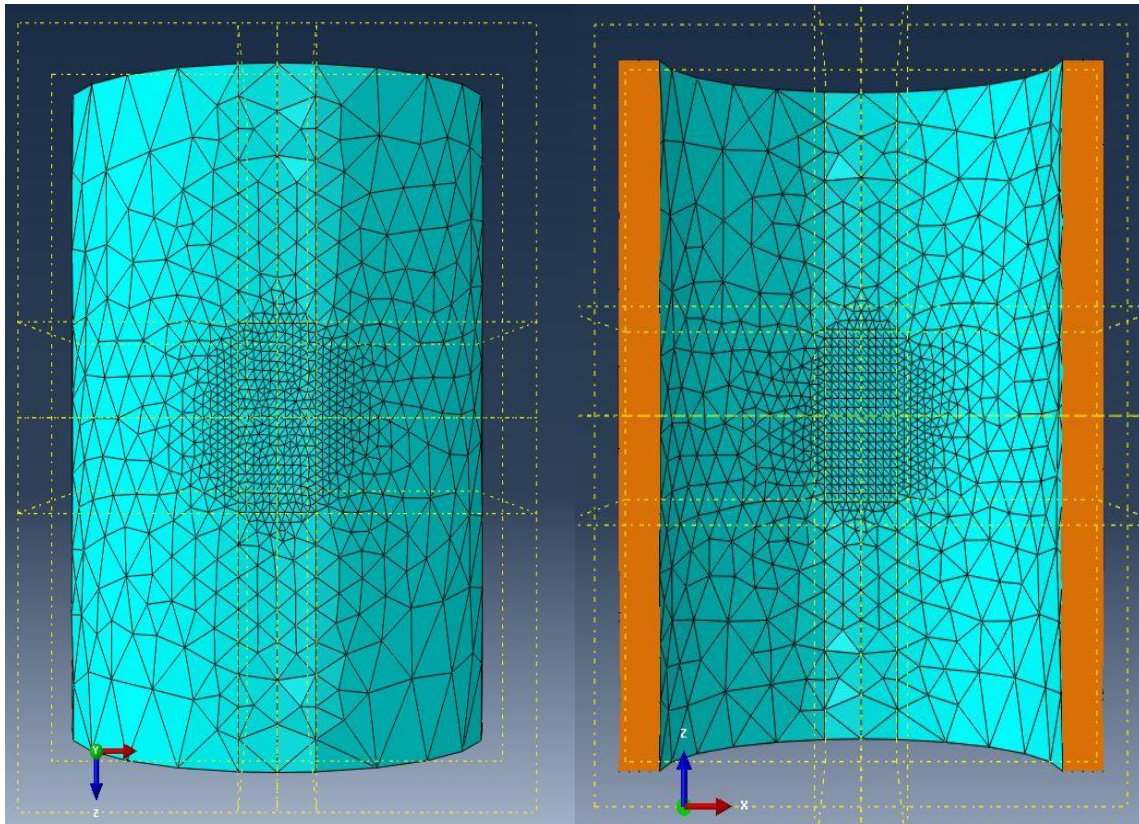
Figure 18: Mesh for pipe section for numerical simulation for creep loading

2.5.3 Material model

The material model for this creep analysis consist of both Cohesive zone properties and also creep properties from literature. The section used is solid section with 3D stress element for the analysis. The properties for the material PA11 are listed below in a Table 5

Table 5: Creep properties of material PA11

S.No	Property of Material	Value
1	Modulus of Elasticity	1700 MPa
2	Poisson's Ratio	0.4
3	Utimate Tensile Strength	68.9 MPa

4	Yield Tensile Strength	44 MPa
5	Max. Principal Stress value for CZM	62 MPa
6	Fracture Energy for CZM	620 KJ/m2
7	Creep Power Law Multiplier	2.45E-7 MPa-n time-1
8	Creep Stress exponent, m	6.5
9	Time order	0
10	Dimensionless time period	70
11	Creep tolerance	0.00001

2.5.4 Loading and boundary condition

The boundary condition is similar to the previous analysis as the basic principle of analysis

remains the same. The loading for this creep analysis is different, as explained earlier the

load is applied over a time period. The full load is applied within the first time period and

then it is maintained at this load for rest of time period. The internal pressure of the pipeline

was maintained as 30 MPa and load due to rock impingement was 15 MPa The region of

loading and boundary conditions are shown Figure 19.
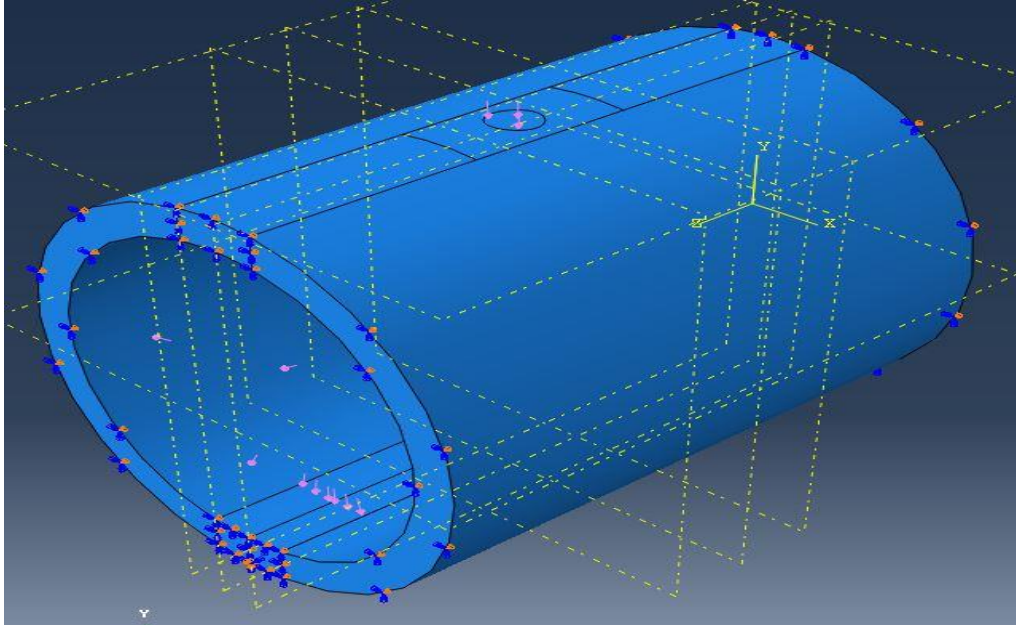
Figure 19: Loading region and Boundary Condition for creep analysis

29

2.5.5 Numerical Simulation result for Creep Analysis

The simulation is similar to that of previous analysis except an additional creep factor is added. In this model, a Visco step is added to do the creep analysis along with the crack domain definition, crack initiation criterion and crack evolution criterion. The important factor of creep analysis is time period. The time period chosen is 70, for which the analysis takes place. The analysis uses initial and minimal time increment as 0.001 and maximum time increment as 1. The tolerance for maximum difference in creep strain increment has been set as 1e-5

In stress based model, the calculation is done to convert the time period obtained from analysis to actual time period. In general, the time period is multiplied with 365 days and 24 hours as the creep constant is in terms of year time unit. The value is then multiplied by a constant (multiplying factor) to get the appropriate time in actual time frame. But by modifying the creep constant it is possible to get direct real-time value from simulation.

In the strain based model, the time period for the simulation can be converted to real-time time by multiplying by same methodology. The constant for creep analysis has been modified such that it gives the real time life of the pipe model. A nominal value of creep constant is to be chosen if it is very small time then the crack propagation is very quick and rapid, thus crack growth cannot be studied in detail whereas if it is very high small crack propagation is very slow, this takes a lot of computational time.

The stress based simulation results are shown in Figure 20. It shows the crack through thickness of the pipe due to creep loading. Since a constant pressure is applied by a rock impingement the creep crack grows as time period increases. The crack simulation shows the initiation of crack within the material and tends to travel along the thickness first and then travels in axial direction.
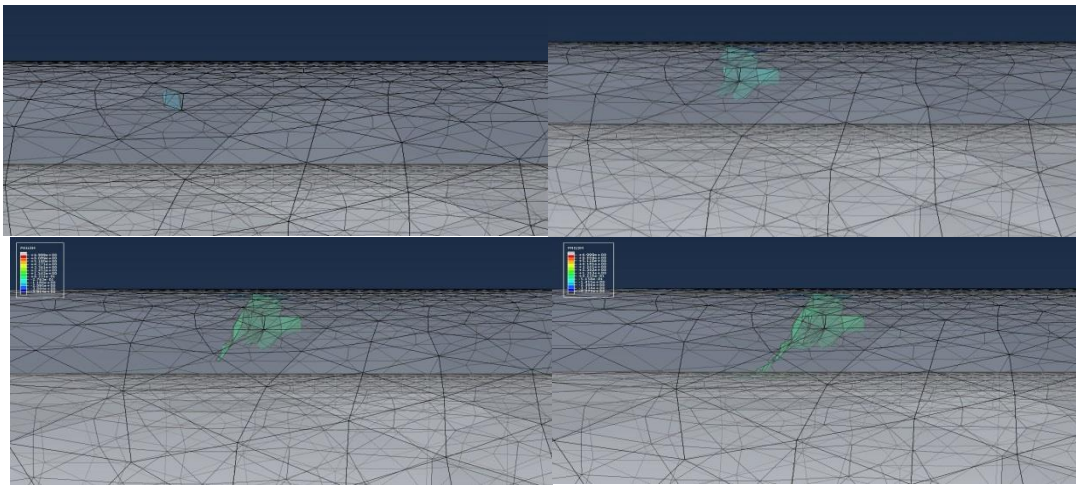
Figure 20: Crack propagation a) initial loading b) t=10.09 c) t=21.98 d) t= 67.58

The strain based simulation results are shown in Figure 21. This simulation is strain based which is representative of the sudden rupture in pipeline due to sharp rock impingement with high loading. This scenario tries to model the hole that we see as effect of sudden rock impingement force.
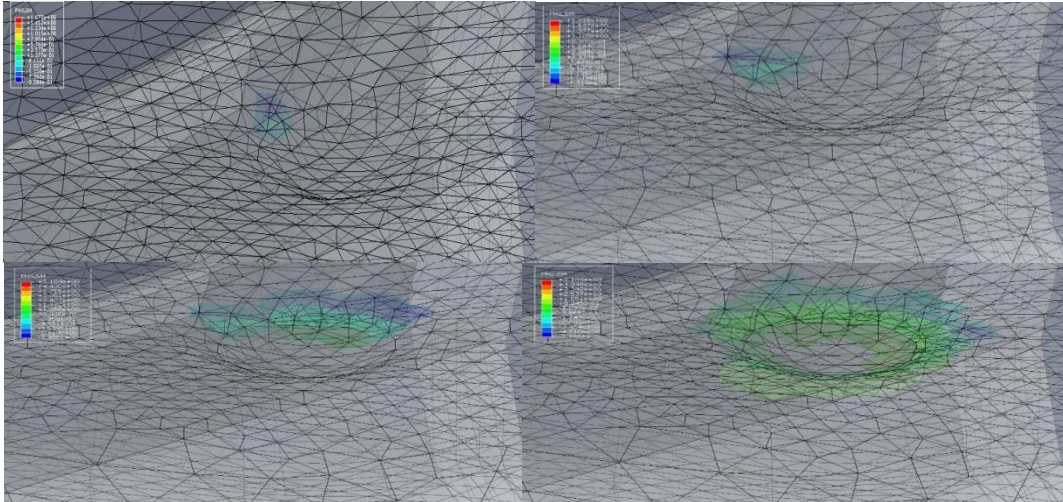
Figure 21: Crack propagation a) initial loading b) t=10.09 c) t=21.98 d) t= 67.58

2.5.6 Design of Experiment for Creep Analysis

Design of experiments are widely used for generating samples that can uniformly cover the whole domain of the explanatory variables [15,16]. In order to accurately capture the system characteristics, a full factorial approach is necessary to investigate all possible combinations. However, it is infeasible to do the full factorial design for high dimensional problems. Latin hypercube design is considered as an optimal method for high dimensional experimental design, because every variable can be represented identically no matter how many samples are selected [17]. In this study, five variables are chosen to investigate the system response based on the numerical investigation. Latin hypercube sampling is employed to generate multiple combinations (i.e. DoE) of design variables. The corresponding system response of each DoE can be computed using the developed finite element model. With these samples, a surrogate model that can capture this system

32

behavior is developed, which makes it possible for probabilistic analysis with respect to each design variable using Bayesian updating.

The Latin hypercube sampling (LHS) is very flexible to be used to create any number of design of experiments. A lot of work has been done on the optimization of Latin hypercube from both optimization algorithm and objective criteria [18,19,20,21]. An experiment design with N point in D dimension is given as $X = [x_1, x_2, \ldots x_N]^T$, in which each column represents a random variable and each row x_i represents a realization. A Latin hypercube design is performed in such a way that each dimension is divided into N equally probable intervals and there is only one sample within each interval. In this study, a criterion with maximizing the minimum distance is implemented to generate Latin hypercube samples. Five random variables are chosen and the corresponding sampling interval for each variable is listed in Table 6.

Table 6: Sampling interval for each random variable

Random variable	Creep constant $(Mpa/Time)$	Fracture Energy (MJ/mm^2)	n	Principal stress (Mpa)	Load (Mpa)
Mean	2.15×10^{-3}	0.620	6.5	51	25
Sampling interval	$[1.15,3.25] \times 10^{-3}$	[0.590, 0.650]	[4.5,8.5]	[46,56]	[15,35]

Using Latin hypercube sampling method, 20 design of experiments are generated which are listed in Table 7. The projection of these samples on 2D plane is shown in Figure 22.
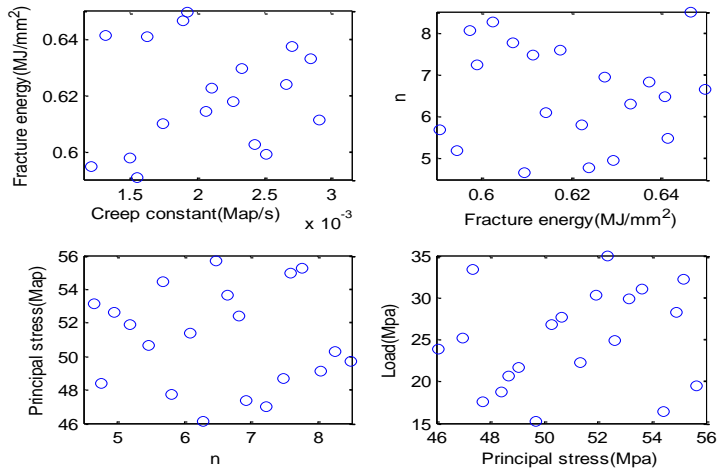
Figure 22: Projection of 20 samples on each 2D plane

For each DoE, the system response such as crack length vs. time curve can be generated using the FEM model developed above. During simulation, some cases may crash because of impractical variable combination is input into the FEM model. By ignoring them, crack growth curves for 8 DoEs are plotted in Figure 23.
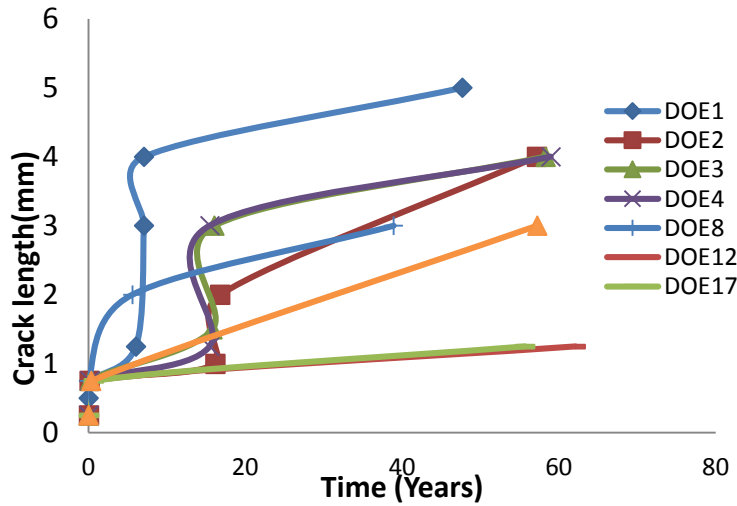
Figure 23: Crack growth curves for 8 DoEs

Table 7: The 20 DoE samples generated using LHS

S.No	Creep Constant A (e-3)	Fracture Energy (e-3)	Stress Order n	Principal Stress	Constant Load
1	1.625218	640.9721293	6.477983	55.67172305	19.40992
2	2.702999	637.4050148	6.831243	52.36128171	34.99771
3	1.930486	649.9379596	6.647187	53.64543485	31.05547
4	2.272673	617.6231862	7.583964	54.93485224	28.20752
5	2.104886	622.4275118	5.812922	47.73800136	17.60067
6	2.509233	599.1437521	7.21698	46.97705494	25.12655
7	1.554258	590.6893212	5.689157	54.44934716	16.3256
8	2.426683	602.5502794	8.248061	50.26553246	26.75763
9	1.744176	609.7218497	4.651164	53.14467682	29.83156
10	1.100611	607.0993455	7.758181	55.1993268	32.13888
11	2.66107	623.9591286	4.76823	48.39754151	18.67336
12	2.911832	611.424453	7.476164	48.69060674	20.67485
13	1.310673	641.3846212	5.474156	50.65042494	27.59983
14	2.0614	614.3088125	6.084406	51.34258566	22.23649
15	2.327245	629.4692397	4.952952	52.59760568	24.83593
16	1.208735	594.5445463	5.18877	51.91098878	30.31171
17	1.009587	627.5343785	6.933126	47.37934714	33.35668
18	1.891855	646.8039981	8.478775	49.70263502	15.17865
19	1.496666	597.6290569	8.048688	49.06588354	21.62145
20	2.842759	633.2133341	6.289175	46.09066581	23.8881

Conceptually, the developed surrogate model should be a time dependent model that has design variables as input parameters. However, it is not that straight forward to get this model in one step. The idea in this study is to fit the crack length with respect to time relationship using power law, and then the effect of design variables is incorporated in the model coefficients using other regression methods. Bearing this idea in mind, the proposed surrogate model is expressed as

$$a = f_1(P)t^{f_2(P)} + f_3(P) \qquad (13)$$

where t is the creep time and $f_1(P) = \alpha_0 + \alpha_i p_i$, $f_2(P) = \beta_0 + \beta_i p_i$, $f_3(P) = \gamma_0 + \gamma_i p_i$, in which p_i represents the design variables. After fitting this model, the true and predicted crack length vs. time data are illustrated in Figure 24. In order to investigate the model sensitivity with respect to each design variable, the crack length variation due to 10% perturbation of each design variable at creep time $t = 10$ years is shown in figure 25.
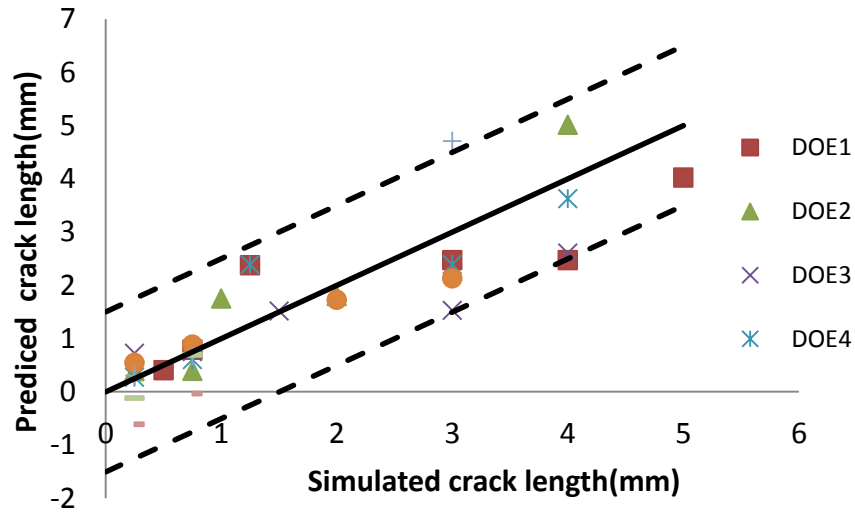
Figure 24: Comparison between true and predicted crack length
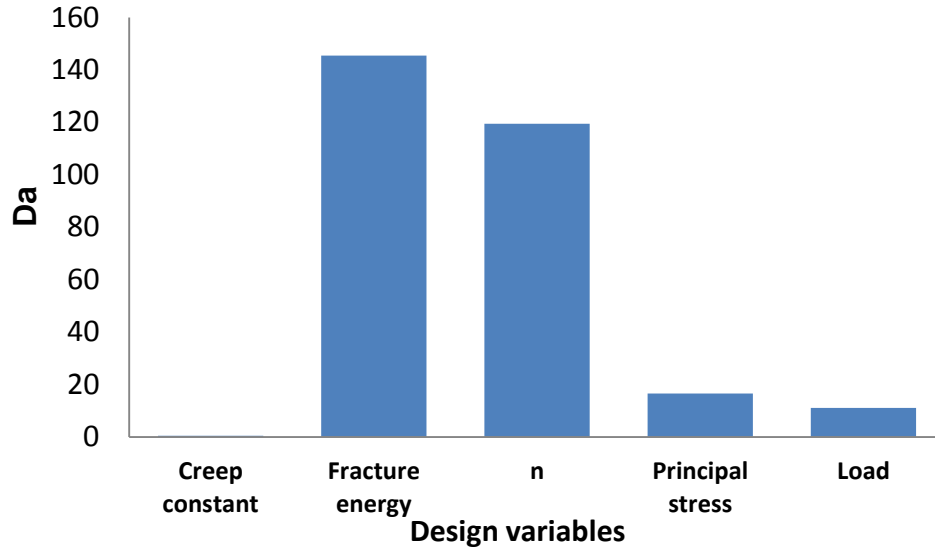
Figure 25: Crack length variation for the perturbation of each design variable

As seen in Figure 24, the proposed model can generally capture the crack growth trend for different DoE samples, but there are relatively large uncertainties embedded in the prediction. Based on this model, it can be concluded that the crack growth is more sensitive to the parameters that control the crack initiation and material creep behavior, which is given as the fracture energy and n in Figure 25.

3. NUMERICAL STUDY ON USING COHESIVE ZONE MODEL FOR COMPOSITE MATERIAL

This section proposes a model that can be used for high cycle fatigue loading for composite material analysis using cohesive zone model. In literature, most of the work on the fatigue damage accumulation for high cycle fatigue follows a cyclic model which consumes more computation time and is theoretically more complex. Hence in this work, an attempt has been made to create a model which can be applied with Abaqus simulation software to address the high cycle fatigue damage using cohesive zone elements.

3.1 Model for fatigue damage accumulation using cohesive zone model

The various models developed in the past have been very successful in addressing low cycle fatigue but there are a very few models for high cycle fatigue loading. There are mainly two methods to account for cyclic loading: loading and unloading pattern and degrading stiffness on a cycle by cycle basis which is mainly used for low cycle fatigue application where there is significant hysteresis, the other method is developing an envelope strategy, where maximum load remains constant (like creep loading) for a particular time period and interface elements are degraded on discrete number of cycles after each time step. The second strategy is the basis for this methodology which has two main requirements: extraction of strain energy release rate from elements within cohesive zone; enabling the interface elements within the cohesive elements to be degraded such that rate of crack advance matches with that given by the Paris law from the corresponding strain energy release rate. The envelope strategy has been adopted for high cycle fatigue damage accumulation prediction in Cohesive zone model using ABAQUS simulation

software. The cohesive zone model has been explained earlier and the stress degradation occurs depending on Damage parameter, D as shown in the equation below

$$\sigma = \sigma_{max} (1 - D) \tag{14}$$

In order to account for the fatigue damage this damage parameter D is divided into two parts, one for static d_s and other for fatigue damage d_f where the damage due to static is defined in abaqus by default as

$$d_s = \frac{\delta - \delta_e}{\delta_f - \delta_e} \tag{15}$$

In this model the factor d_f is calculated and added to damage due to static to form a total damage accumulation D_{tot}, which modifies the strength degradation law as follows

$$\sigma = \sigma_{max} (1 - D_{tot}) \tag{16}$$

$$D_{tot} = d_s + d_f \tag{17}$$

The main user inputs required for this analysis is Number of cycles N and R-ratio which is a ratio of minimum load to maximum load (P_{min}/P_{max}). The main assumptions for the model are it is a constant amplitude loading and tension-tension fatigue loading. Thus the maximum strain energy release rate in each fatigue cycle G_{max} can be converted to change in strain energy release rate ΔG by using the equation below

$$\Delta G = (1 - R^2)G_{max} \tag{18}$$

In our model the maximum strain energy release rate can be obtained from the cohesive elements from the traction-separation response directly adjacent to numerical crack tip

In order to calculate the mixed mode crack growth rate the Paris law that equates crack growth rate and stress intensity factor is used

$$\frac{\partial a}{\partial N} = C(\Delta K)^m \tag{19}$$

By using modified Blanco's model which is a modified version of Paris law we can write the equation as

$$\frac{\partial a}{\partial N} = C(\Delta G)^m \tag{20}$$

Where ΔG is change in strain energy rate and C and m are material constants which can be determined from the equations below, where C_I, C_{II}, C_m and m_I, m_{II}, m_m are mode I, mode II and mixed mode constants.

$$\log C = \log C_I + \left(\frac{G_{II}}{G_T}\right) \log C_m + \left(\frac{G_{II}}{G_T}\right)^2 \log \frac{C_{II}}{C_m C_I} \tag{21}$$

$$m = m_I + \left(\frac{G_{II}}{G_T}\right) m_m + (m_{II} - m_I - m_m) \left(\frac{G_{II}}{G_T}\right)^2 \tag{22}$$

G_T, G_I and G_{II} are the total, mode I and mode II strain energy rates which are obtained from the traction displacement response of cohesive elements. The fatigue damage degrades according to the strain energy release rate extracted from the traction separation law. The total rate of fatigue damage accumulation is dependent on the crack growth $\frac{\partial a}{\partial N}$ and interface element crack length L_e. The relationship can be written as

$$\frac{\partial a}{\partial N} = \frac{L_e}{\Delta N_f} \tag{23}$$

But our requirement for the propagation of crack is damage accumulated due to fatigue in each cycle $\frac{\partial d_f}{\partial N}$ which can be derived on the basis of failure of the element due to static and fatigue damage as following

$$\frac{\partial d_f}{\partial N} = \frac{(1-d_s)}{\Delta N_f} \tag{24}$$

$$\frac{\partial d_f}{\partial N} = \frac{(1-d_s)}{L_e} \left(\frac{\partial a}{\partial N}\right) \tag{25}$$

Thus, the damage due to fatigue is updated every time period and this can be expressed by using the equation (26) where ΔN is change in number of cycle in that particular time period

$$d_{f,t+\Delta t} = d_{f,t} + \Delta N \left(\frac{\partial d_f}{\partial N}\right) \tag{26}$$

3.2 Modeling for high cycle fatigue damage in Double Cantilever Beam model

The model to account for fatigue damage accumulation is used to analyze the delamination in composite based Double Cantilever Beam in order to verify its accuracy with experimental data available in Literature.

3.2.1 Modelling Technique

The Double Cantilever Beam model has been used for mode I fatigue testing for a very long time and gives most accurate values which is used for verifying the model proposed for high cycle fatigue damage accumulation in numerical simulation. In this model two beams of length 120 mm and breath of 20 mm are attached to each other with a cohesive element layer. The cohesive elements in between the layers serve as the interface elements in our fatigue damage model. The cohesive layer serves as the adhesive layer (epoxy/polyamide) in case of a composite layup and hence this numerical simulation can be viewed as delamination of composite material. The specimen used for analysis is shown below.

h=3.1 mm

a_o = 35 mm

L=120 mm

b=20 mm
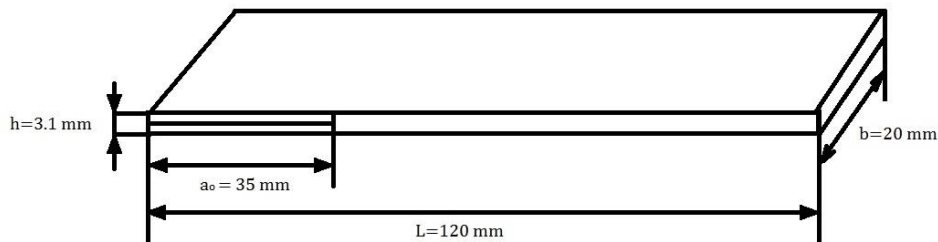
Figure 26: Specimen Geometry

The model is built for 8-node rectangular cohesive 3D element and the modification of damage criterion is incorporated in UEL which has to be used along with the ABAQUS input file.

3.2.2 Material Model

The materials used in this analysis are divided into two categories: Laminate properties and interface properties. There are two element types used - shell element for the laminate properties and cohesive element for interface properties. Similarly, the section assignment for laminate element is composite shell and for interface element it is cohesive section. The properties for both elements are listed below in Table 8

Table 8: Properties of DCB simulation

S.No	Property of Material	Value
Laminate Properties		
1	E11	120000 MPa
2	E22 = E33	10500 MPa
3	G12=G13	5250 MPa
4	G23	1500 MPa
5	$\nu 12 = \nu 13$	0.3
6	$\nu 23$	0.51
Interface Properties		
7	G_{IC}	0.26 N/mm
8	G_{IIC}	1.002 N/mm
9	σ_{Imax}	30 MPa
10	σ_{IImax}	60 MPa
11	K_I	10^5 N/mm^3
12	K_{II}	10^5 N/mm^3
13	R	0.1
14	N	1000 cycles
Blanco's Paris law model constants		
15	C_I	2.1 (mm/cycle) (N/mm)$^{-m}$
16	C_{II}	0.12 (mm/cycle) (N/mm)$^{-m}$

43

17	C_m	436600 (mm/cycle) (N/mm)$^{-m}$
18	m_I	5.09
19	m_{II}	4.38
20	m_m	5.48

3.2.3 Loading and Boundary condition

The loading for this model is an extension of creep model where the load was maintained constant after reaching the maximum load, thus during this time period the fatigue law is activated in this region, during which the damage due to fatigue d_f term is calculated and strength degradation occurs. The loading for the high cycle fatigue damage is shown below in Figure 27.
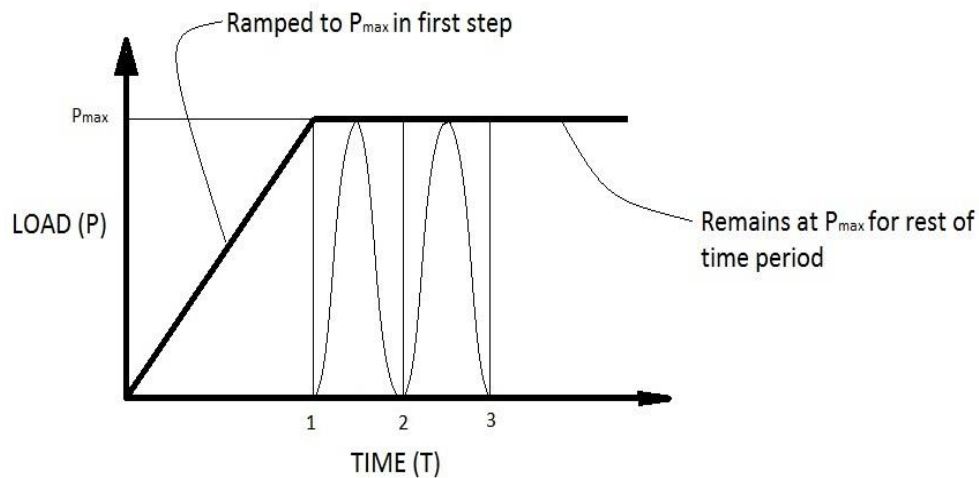
Figure 27: Schematic diagram of loading in fatigue cycle

The boundary condition for DCB model is fixing one end of the Cantilever beam and creating a pre-crack of 35 mm on the other side, and the application of load is on width of the beam on the side of pre-crack.

3.2.4 Numerical simulation results

The simulation for Double Cantilever Beam requires two inputs along with the model that has been prepared, the input file in ABAQUS and UEL file which has modified COH3D8 element for accounting the fatigue damage accumulation. The crack propagation at different time period which actually indicates number of cycle under high cycle fatigue is shown in

Figure 28 below
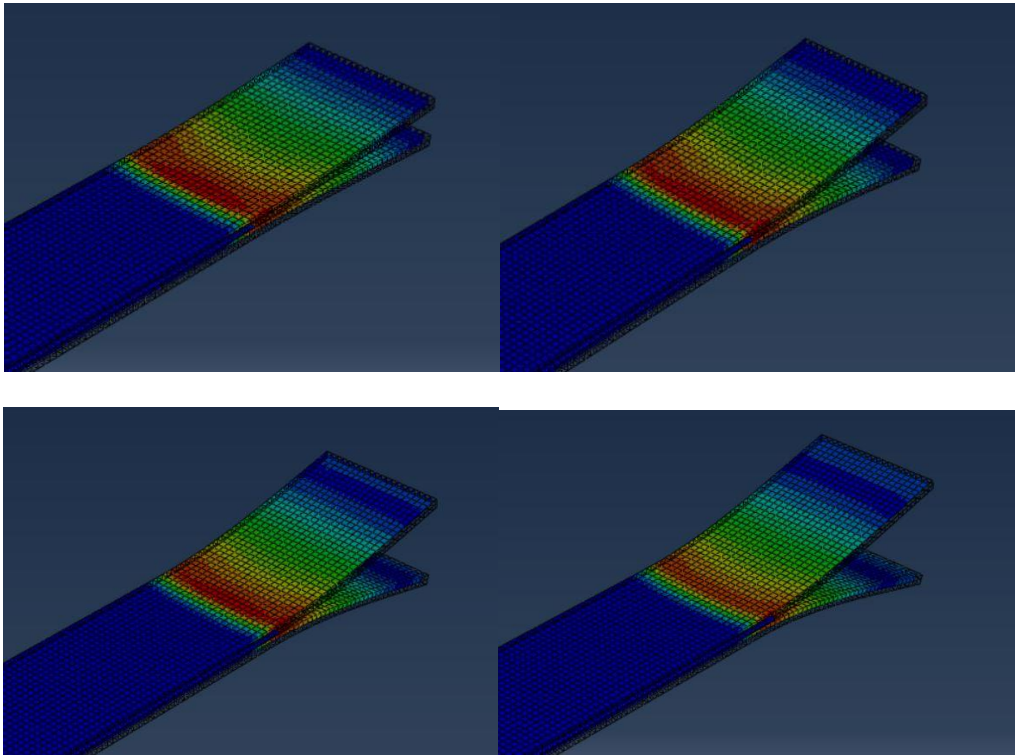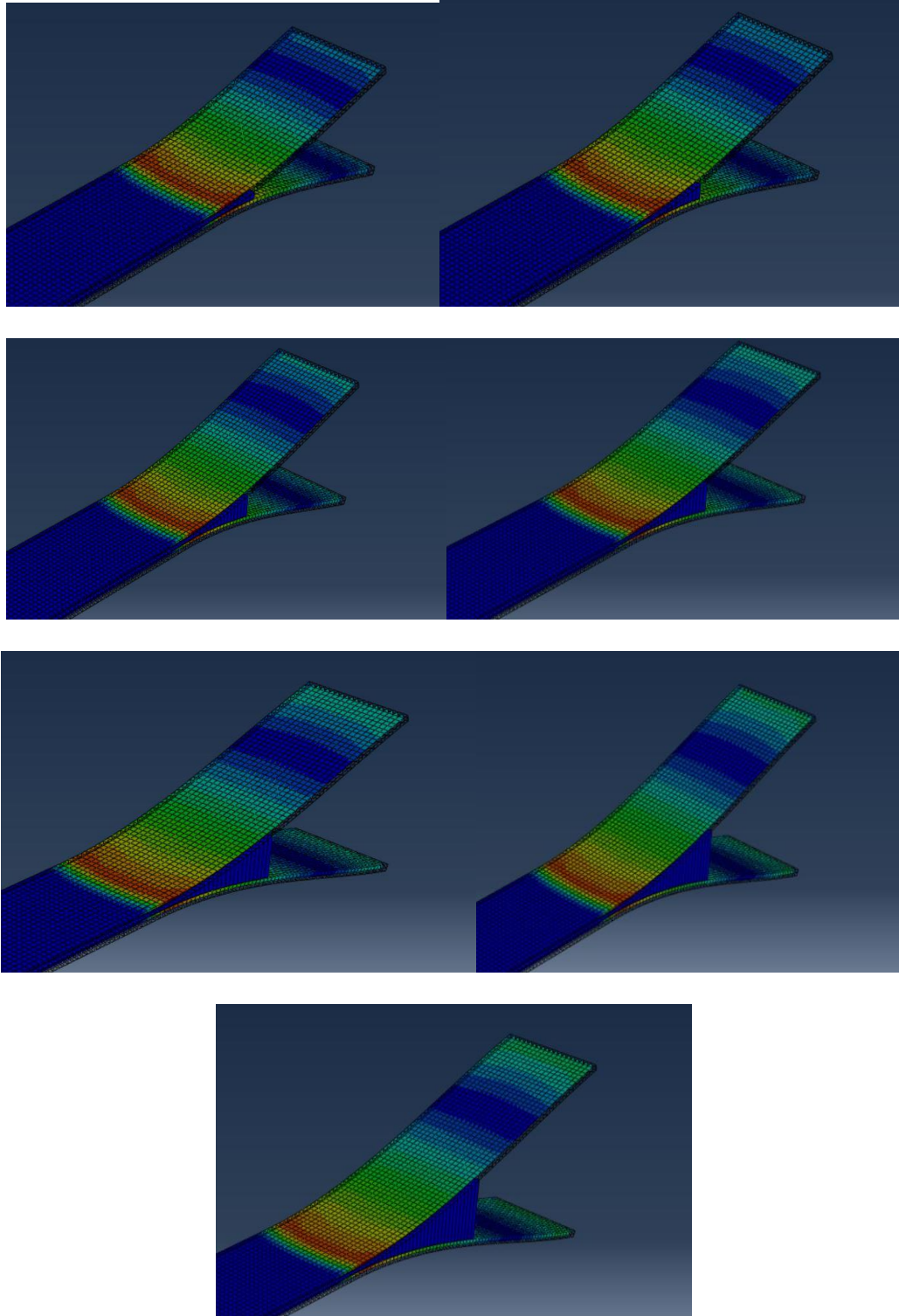
Figure 28: Crack propagation at different time period for DCB specimen (from t=1 to 11)

It clearly indicates that the crack growth is rapid during the last time period which indicates high number of cycles, in order to visualize this effect two plots of $\frac{\partial a}{\partial N}$ vs G_I/G_{IC} which has maximum stress values of 30 MPa and 60 MPa is shown below in Figure 29.
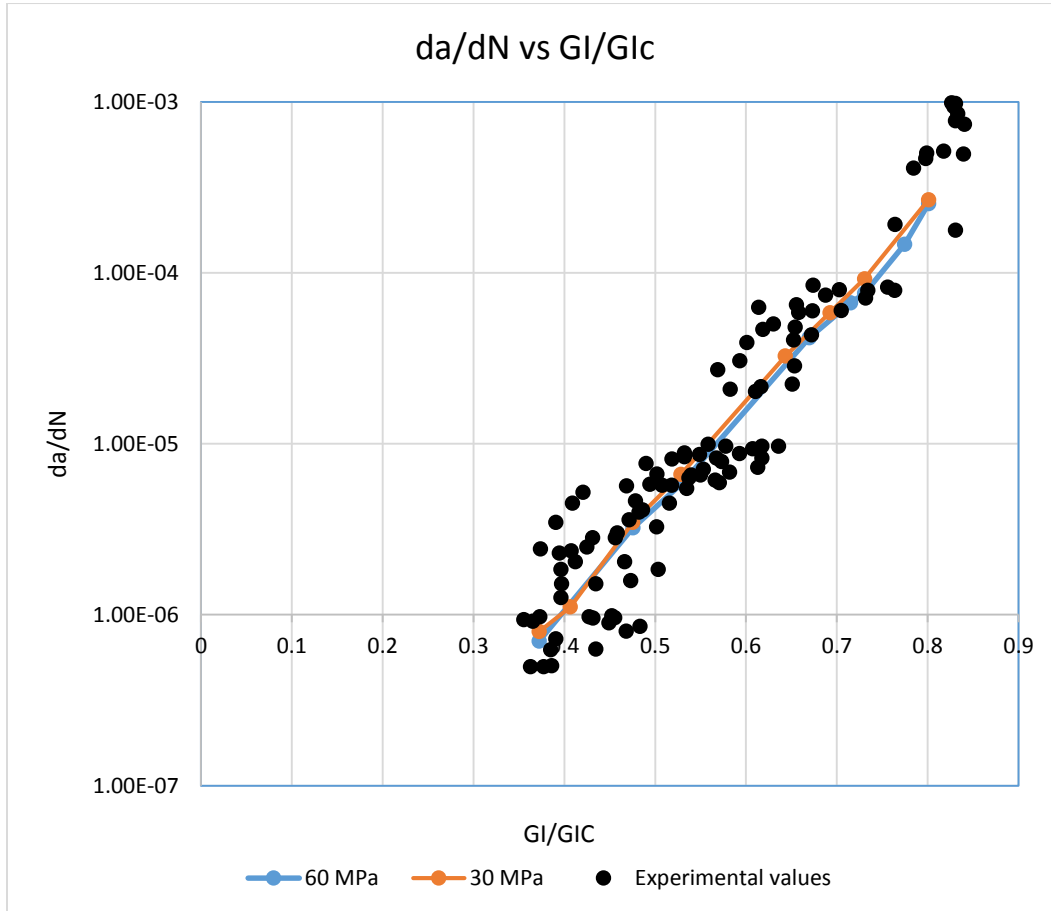
Figure 29: Rate of crack growth per cycle for DCB specimen

Also, the crack growth versus number of cycles plot has been shown below in Figure 30 to indicate the crack growth in high cycle fatigue damage
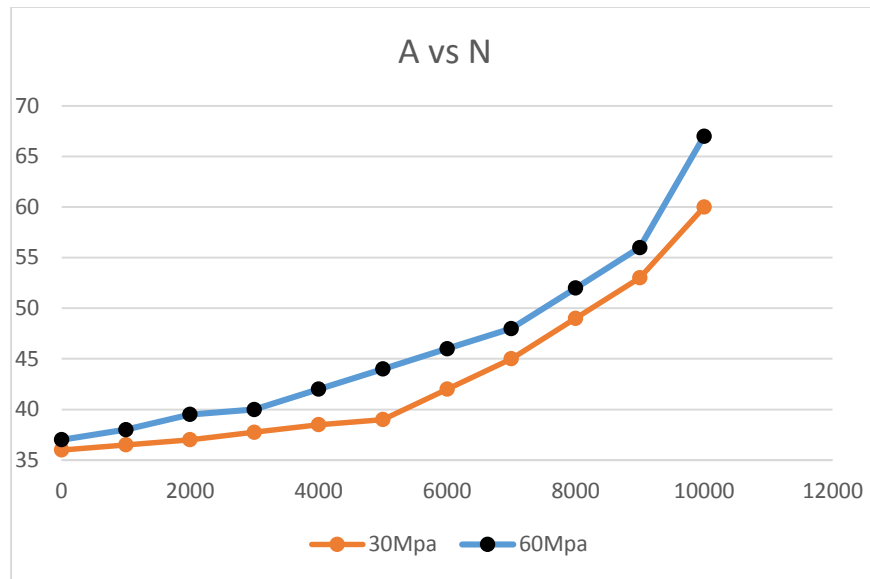
Figure 30: Plot of crack growth vs number of cycles for DCB specimen

3.3 Application of High cycle fatigue model for pipeline system

3.3.1   Background of the problem

The model for fatigue damage accumulation is used for real-time application in Cured-IN pipe structure [22]. There are special types of pipelines used for transport of high pressure natural gas called the Cured-In-Plane Liner pipelines, which has a liner material and a metallic pipeline layer. In general, the inner material is composite and outer material is metal. In these pipelines the pressure of natural gas is very high and when they are depressurized for maintenance or emergency repair, there is a void formed between the liner material and the pipe as shown in Figure 31.
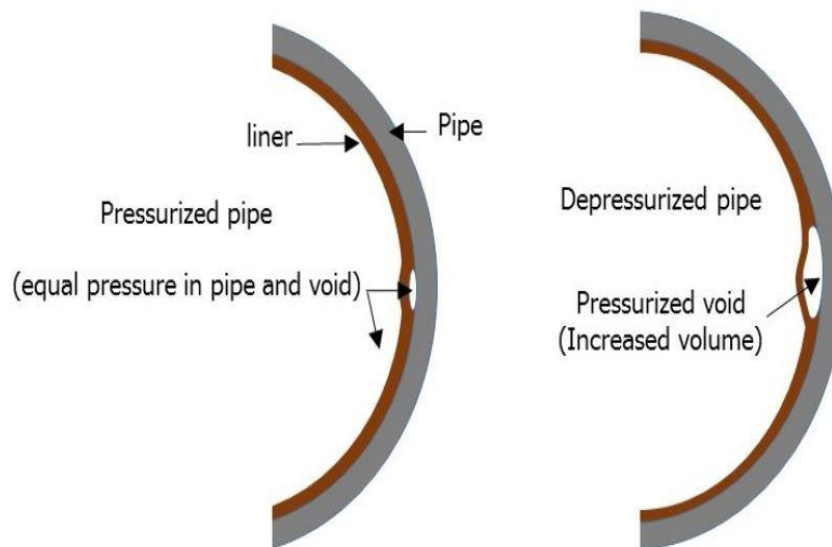
Figure 31[22]: Pipe and liner material with pressurized void

Once the void is formed, the variation in the pressure in the pipeline is going to induce a cyclic loading which creates a fatigue damage in the piping system. Thus the high cycle

fatigue damage model can be applied to this piping system to study the effect of fatigue damage on the Cured-In-plane Liner pipeline.

3.3.2   Numerical simulation and results

The numerical simulation for Cured-In-plane pipeline system has been done using ABAQUS with the fatigue model developed for high cycle fatigue damage. The simulation model prepared consists of three layers: outer layer which is metallic pipe, inner layer which is liner material and a cohesive element layer in between to address the fatigue damage due to high cycle loading. In this model, only a quarter of pipe model is developed as the region where pressurized void is formed is the region under study. The reason for a quarter model is that it uses lesser number of elements and hence the computation time and cost would reduce. The modeling is done in two stages were the pipe is first modelled with a dimension of outer diameter of 120 mm and inner diameter of 100 and then the model is cut in between to create cohesive layer with a thickness of 5 mm. The model created is shown below in Figure 32.
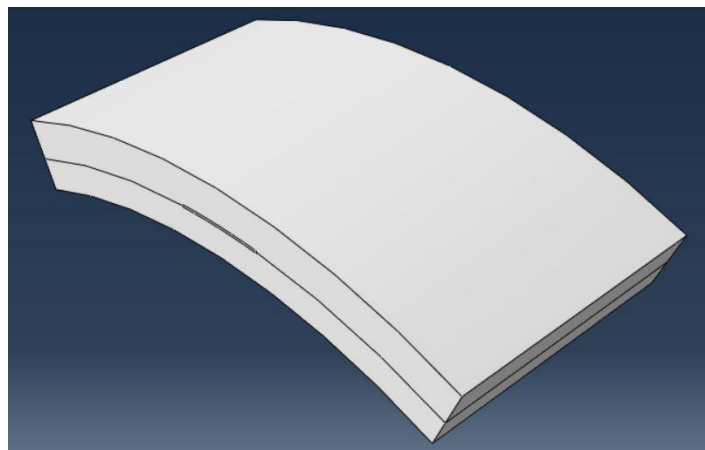
Figure 32: Model for Cured-In-Plane pipeline system

3.3.2.1 Material Model

The material model consists of three section: metallic part, cohesive section and liner material. The corresponding section assignment for each part is solid section, cohesive section and shell composite. The properties associated with the model is listed below in the table

Table 9: Properties of various material in model

S.No	Property of Material	Value
Metal properties		
1	Young's Modulus	20500
2	Poisson's Ratio	0.4
Laminate Properties		
3	E11	120000 MPa
4	E22 = E33	10500 MPa
5	G12=G13	5250 MPa
6	G23	1500 MPa
7	$\nu12 = \nu13$	0.3
8	$\nu23$	0.51
Interface Properties		
9	G_{IC}	0.26 N/mm
10	G_{IIC}	1.002 N/mm
11	σ_{Imax}	30 MPa
12	σ_{IImax}	60 MPa
13	K_I	10^5 N/mm^3
14	K_{II}	10^5 N/mm^3
15	R	0.1
16	N	1000 cycles
Blanco's Paris law model constants		
17	C_I	2.1 (mm/cycle) (N/mm)$^{-m}$
18	C_{II}	0.12 (mm/cycle) (N/mm)$^{-m}$
19	C_m	436600 (mm/cycle) (N/mm)$^{-m}$
20	m_I	5.09
21	m_{II}	4.38
22	m_m	5.48

3.3.2.2 Loading and boundary condition

The loading is given at the cohesive layer between the liner material and the pipeline in order to study the fatigue damage due to void creation. The edges of the model have a fixed boundary conditions in order to have a stability in analysis. The internal pressure of the pipeline was maintained as 100 MPa and load at the layer was 50 MPa. The figure below shows the boundary condition and loading given to the model.
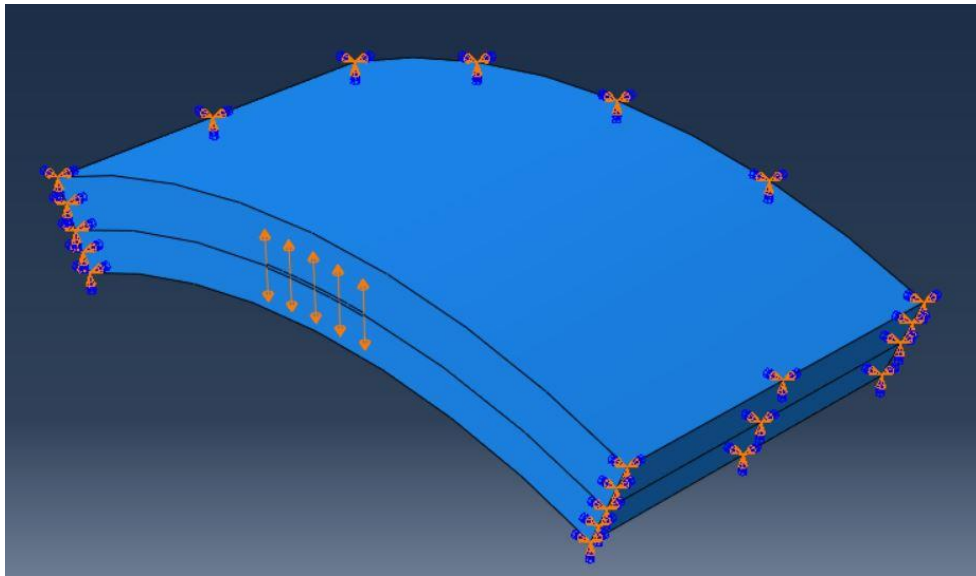
Figure 33: Boundary condition and Loading

3.3.2.3 Simulation results for analysis

The simulation results show the void formation due to fatigue and its enlargement and increment in void pressure over time due to cycle loading. The fatigue damage is restricted to the layer because only cohesive element has fatigue damage accumulation term. Figure 34 final stage of crack propagation with respect to time period.
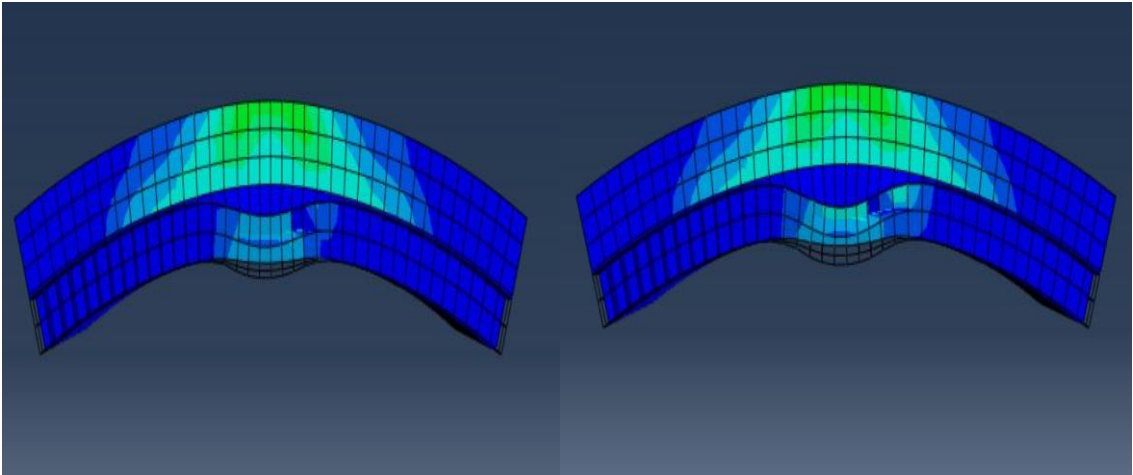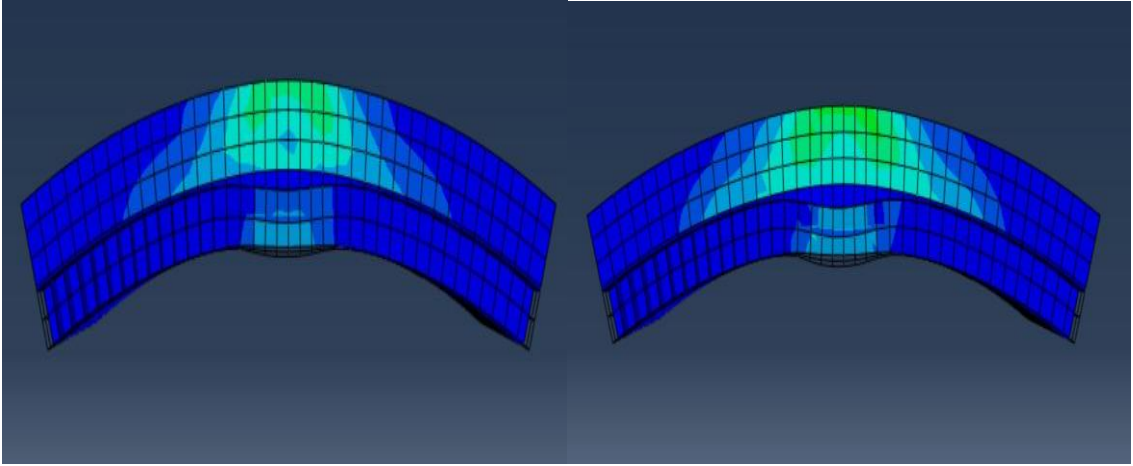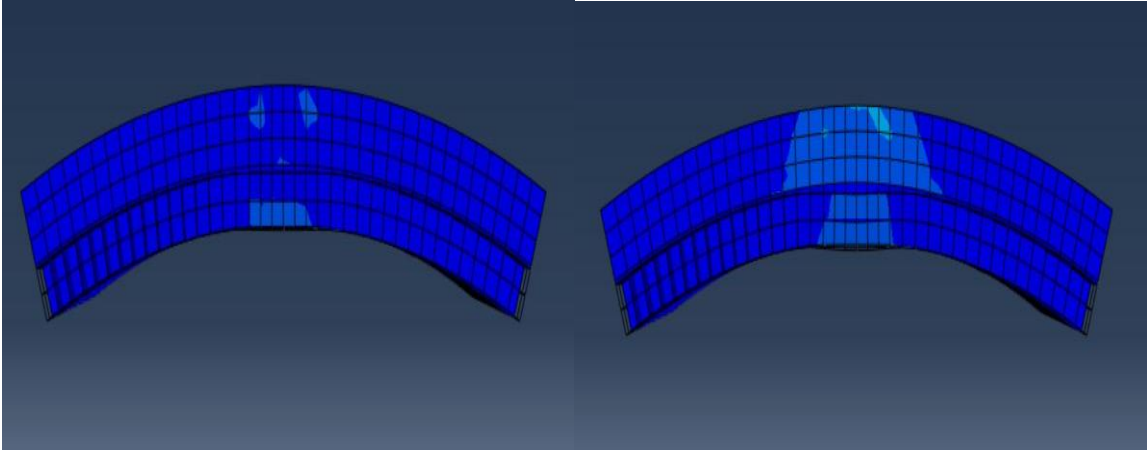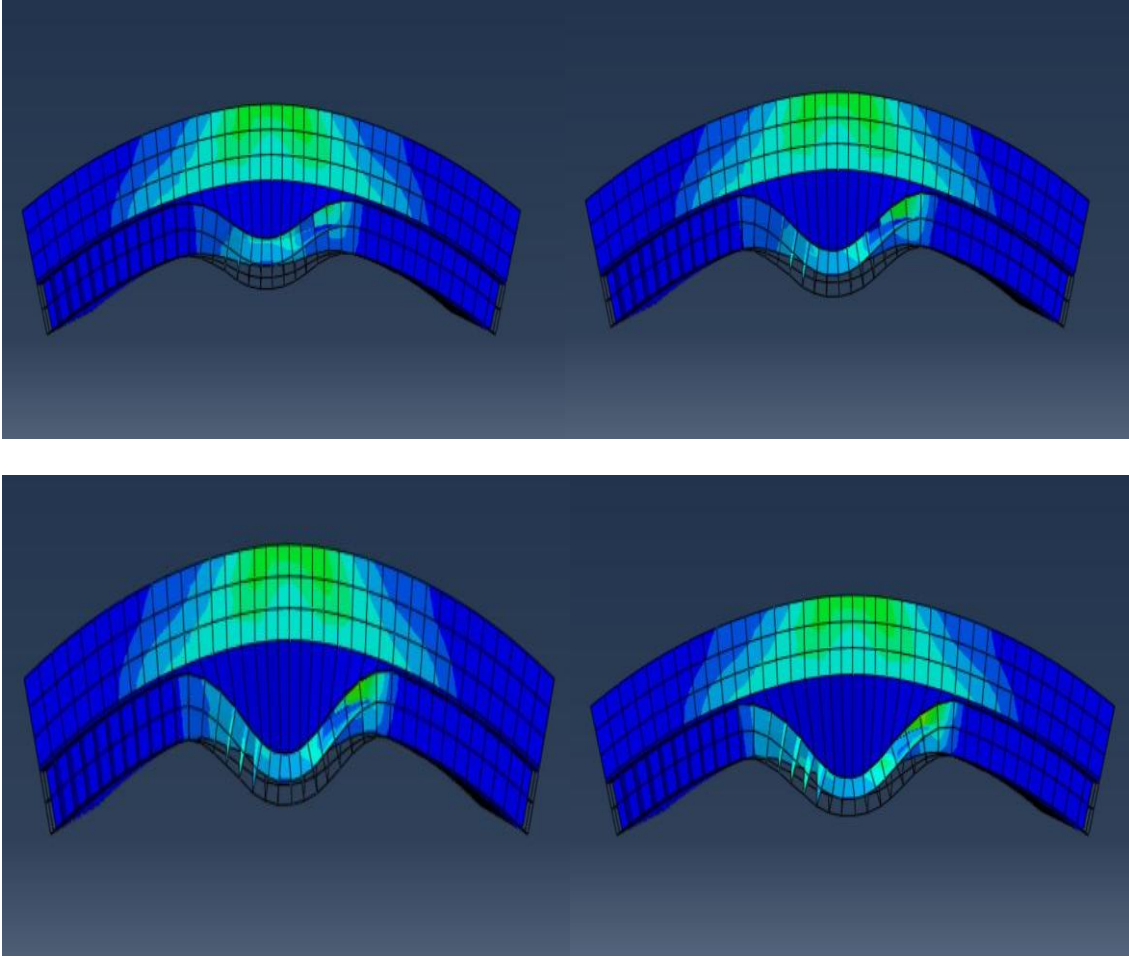
Figure 34: Final Stage of crack propagation with respect to time period

4. SUMMARY

In conclusion, cohesive zone model has been studied for various loading conditions and analysis. In static loading case we were able to get a desirable result by using CZM in conjunction with XFEM which overcomes the disadvantage of cohesive zone model of indication of crack direction. It also shows this methodology applies well for failure analysis as both XFEM and CZM compliments each other in reducing their disadvantages. The real time problem of pipeline crack propagation due to rock impingement is also done using this methodology. Also, parametric study was conducted which gives a clear illustration of effect of internal diameter, pipe thickness and cavity on crack initiation and propagation.

In the next part, a time dependent creep loading is used where the load is held for particular time period in and the crack propagation is increased. The creep model developed using CZM and XFEM is tested by using numerical simulation. Also, Design of experiment was done using Latin cube method to define a model for life prediction. This gave a good insight on the crack propagation that occurs over time with minimal variation in temperature factor. The model was then used for simulation of crack propagation in pipe impingement problem to obtain a realistic life prediction of an in service gas pipeline. The results were satisfying and were quite a good match with the literature values.

In the third chapter, a model was developed to account for fatigue damage accumulation where a factor was added to existing damage due to static. The loading was similar to that of creep model were the maximum load was maintained and the time period is increased

but here with the increase in every time period there is particular number of cyclic loading associated with it thus leading to fatigue damage. This fatigue damage factor is determined and added to existing static analysis. Thus the model account for both static and fatigue damage.

There is a lot of scope for future work in this field such as addition of XFEM in fatigue damage accumulation which will help to determine the crack propagation without crack path definition. Moreover, the mixed mode criterion can be altered to see its applicability with other models as well.

REFERENCES

[1] Zavattieri, Pablo D. November 2006. "Modeling of Crack Propagation in Thin-Walled Structures Using a Cohesive Model for Shell Elements." Journal of Applied Mechanics Vol. 73 pp. 948-958.

[2] Park K, Paulino GH. Cohesive Zone Models: A Critical Review of Traction-Separation Relationships Across Fracture Surfaces. ASME. Appl. Mech. Rev.2013;64(6)

[3] Ebrahimi S.H., Mohammadi S., Asadpoure A. "An Extended Finite Element (XFEM) Approach for Crack Analysis in Composite Media" International Journal of Civil Engineering. Vol. 6, No. 3, September 2008.

[4] Saleh Yazdani, Wilhelm J.H. Rust, Peter Wriggers. "An XFEM approach for modelling delamination in composite laminates" Composite Structures 135 (2016) 353–364.

[5] J.L. Bouvard, J.L. Chaboche, F. Feyel, F. Gallerneau. "A cohesive zone model for fatigue and creep–fatigue crack growth in single crystal superalloys" International Journal of Fatigue 31 (2009) 868–879.

[6] Kyungmok Kim. "Creep–rupture model of aluminum alloys: Cohesive zone approach" Proc IMechE Part C: J Mechanical Engineering Science 2015, Vol. 229(8) 1343–1347.

[7] Pouria Hajikarimi, Soheil Mohammadi, Sassan Aflaki. "Two dimensional creep analysis of a linear cracked viscoelastic medium using the extended finite element method" European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012) Vienna, Austria, September 10-14, 2012.

[8] Yangjian Xu; Huang Yuan, "On damage accumulations in the cyclic cohesive zone model for XFEM analysis of mixed-mode fatigue crack growth" Computational Materials Science 46 (2009) 579–585.

[9] Paul W. Harper; Stephen R. Hallett, "A fatigue degradation law for cohesive interface elements – Development and application to composite materials" International Journal of Fatigue 32 (2010) 1774–1787.

[10] Lucien Laiarinandrasana, Guillaume Boisot, C. Fond, Gilles Hochstetter. "Toughness improvement of polyamide 11 assessed via quasistatic tensile tests on notched round bars." ed. D.Klingbeil, M. Vormwald, K.G. Eulitz. Fracture of materials and structures from micro to macro scale - ECF 18, Aug 2010, Dresden, Germany. DVM, 8 p., 2010.

[11] Razavi-Nouri, M., Creep and stress relaxation behavior of polypropylene, metallocene-prepared polyethylene and their blends. Iran. J. Chem. Eng, 2012. 9: p. 60-68.

[12] Osswald, T., Understanding Polymer Processing. Editorial Hanser, USA, 2011: p. 22-23.

[13] McKenna, G.B., On the physics required for prediction of long term performance of polymers and their composites. JOURNAL OF RESEARCH-NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, 1994. 99: p. 169-169.

[14] Meyers, M.A. and K.K. Chawla, Mechanical behavior of materials. Vol. 2. 2009: Cambridge university press Cambridge.

[15] Box, G.E.P. and N.R. Draper, Empirical Model Building and Response Surfaces. John Wiley & Sons, New York, NY, 1987.

[16] Myers, R.H. and D.C. Montgomery, Response surface methodology:process and product optimization using designed experiments. Wiley, New York, 1995.

[17] McKay, M.D., R.J. Beckman, and W.J. Conover, A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. Technometrics (American Statistical Association), 1979. 21(2): p. 239-245.

[18] Audze, P. and V. Eglajs, New approach for planning out of experiments. Problems of Dynamics and Strengths, 1977. 35: p. 104-107.

[19] Park, J.S., Optimal Latin-hypercube designs for computer experiments. Journal of Statistical Planning and Inference, 1994. 39: p. 95-111.

[20] Ye, K.Q. and W. Li, Algorithmic construction of optimal symmetric Latin hypercube designs. Journal of Statistical Planning and Inference, 2000. 90(145-159).

[21] Jin, R., W. Chen, and A. Sudjianto, An efficient algorithm for constructing optimal design of computer experiments. Journal of Statistical Planning and Inference, 2005. 134(268-287).

[22] Khalid Farrag "Evaluation of Structural Liners for the Rehabilitation of Liquid and Natural Gas Piping Systems" Final report from Gas technology Institute.

[23] Daniel W. Spring, Glaucio H. Paulino "A growing library of three-dimensional cohesive elements for use in ABAQUS" Enginering Fracture Mechanics 126 (2014) 190-216.

[24] Abaqus documentation 6.14 "Abaqus/CAE User's Guide"

APPENDIX A

FORTRAN CODE FOR FATIGUE DAMAGE ACCUMULATION

```fortran
      SUBROUTINE UEL (RHS, AMATRX, SVARS, ENERGY, NDOFEL, NRHS,
     NSVARS,PROPS, NPROPS, COORDS, MCRD, NNODE, U, DU, V, A, JTYPE,
     TIME,DTIME, KSTEP, KINC, JELEM, PARAMS, NDLOAD, JDLTYP, ADLMAG,
     PREDEF, NPREDF, LFLAGS, MLVARX, DDLMAG, MDLOAD, PNEWDT, JPROPS,
     NJPROP, PERIOD)
c
      INCLUDE 'ABA_PARAM.INC'
c
      DIMENSION RHS(MLVARX,*),AMATRX(NDOFEL,NDOFEL),PROPS(*),
     SVARS(*),ENERGY(8),COORDS(MCRD,NNODE),U(NDOFEL),
     DU(MLVARX,*),V(NDOFEL),A(NDOFEL),TIME(2),PARAMS(*),
     JDLTYP(MDLOAD,*),ADLMAG(MDLOAD,*),DDLMAG(MDLOAD,*),
     PREDEF(2,NPREDF,NNODE),LFLAGS(*),JPROPS(*)
c
      DIMENSION ds1(4),ds2(4),dn(4),Trac(MCRD,NRHS),
     Trac_Jacob(MCRD,MCRD),R(MCRD,MCRD),coord_l(MCRD,NNODE),
     GP_coord(2),sf(4),B(MCRD,NDOFEL),co_de_m(3,4),
     B_t(NDOFEL,MCRD), Transformation_M(NDOFEL,NDOFEL),
     Transformation_M_T(NDOFEL,NDOFEL),temp1(MCRD,NDOFEL)
c
      DIMENSION stiff_l(NDOFEL,NDOFEL),temp2(NDOFEL,NDOFEL),
     stiff_g(NDOFEL,NDOFEL),residual_l(NDOFEL,NRHS),
     residual_g(NDOFEL,NRHS),aJacob_M(2,3),delu_loc_gp(mcrd),
     co_de(mcrd,nnode)
c
      DOUBLE PRECISION G_fn, G_ft, f_tn, f_tt, alpha, beta, rn, rt,
     alphaV, betaV, alphaP, p_m, p_n, deln, delt, tmax, pmax, opn,
     opt, del_e, del_f, D_st, D_fat

c
c Define Inputs
c
      G_fn=PROPS(1)
      G_ft=PROPS(2)
      f_tn=PROPS(3)
      f_tt=PROPS(4)
      alpha=PROPS(5)
```

61

```fortran
      beta=PROPS(6)
      rn=PROPS(7)
      rt=PROPS(8)
      N = PROPS(9)
      R = PROPS(10)
      C_I = PROPS(11)
          m_I = PROPS(12)
          del_e = PROPS(13)
      alphaV=1
      betaV=1
      alphaP=1
      GP_n=4d0
c
c Initialize Matrices and Vectors
c
      call k_vector_zero(ds1,4)
      call k_vector_zero(ds2,4)
      call k_vector_zero(dn,4)
      call k_matrix_zero(Trac,mcrd,nrhs)
      call k_matrix_zero(Trac_Jacob,mcrd,mcrd)
      call k_matrix_zero(R,mcrd,mcrd)
      call k_matrix_zero(coord_l,mcrd,nnode)
      call k_vector_zero(GP_coord,2)
      call k_vector_zero(sf,4)
      call k_matrix_zero(Transformation_M,ndofel,ndofel)
      call k_matrix_zero(Transformation_M_T,ndofel,ndofel)
      call k_matrix_zero(B,mcrd,ndofel)
      call k_matrix_zero(B_t,ndofel,mcrd)
      call k_matrix_zero(temp1,mcrd,ndofel)
      call k_matrix_zero(stiff_l,ndofel,ndofel)
      call k_matrix_zero(temp2,ndofel,ndofel)
      call k_matrix_zero(stiff_g,ndofel,ndofel)
      call k_matrix_zero(residual_l,ndofel,nrhs)
      call k_matrix_zero(residual_g,ndofel,nrhs)
      call k_matrix_zero(aJacob_M,2,3)
      call k_matrix_zero(rhs,ndofel,nrhs)
      call k_matrix_zero(amatrx,ndofel,ndofel)
      call k_matrix_zero(co_de,mcrd,nnode)
```

```fortran
      a_Jacob=0.d0
c
c Determine Inputs to Cohesive Model
c
      p_m=(alpha*(alpha-1.0)*rn**2.0)/(1.0-alpha*rn**2.0)
      p_n=(beta*(beta-1.0)*rt**2.0)/(1.0-beta*rt**2.0)
      deln=(G_fn/f_tn)*alpha*rn*(1.0-rn)**(alpha-1.0)
     &*((alpha/p_m)+1.0)*((alpha/p_m)*rn+1.0)**(p_m-1.0)
      del_f=(G_fn/f_tn)*rn*(1)**(m_I-1)
      da/dN = C_I*(G_fn)**m_I
      D_st=(deln - del_e)/(del_e - del_f)
      N=N+1000
      D_fat=D_fat+ N*((1-D_st)/1.25)*da/dN
      D_tot = D_fat + D_st
      f_tn= f_tn*(1-D_tot)
c
c Do local computations
c
      do i = 1, mcrd
        do j = 1, nnode
          co_de(i,j)=coords(i,j)+U(3.0*(j-1.0)+i)
        end do
      end do
c
c Do Calculations at Gauss Points
c
      do i = 1, GP_n
c
      call k_matrix_zero(aJacob_M,2,3)
c
      gpt = i
c
      call k_local_coordinates(gpt,co_de,R,coord_l,Transformation_M,
     & Transformation_M_T,a_Jacob,aJacob_M,coords,u,ndofel,nnode,
     & mcrd, SVARS)


c
c Compute shear and normal local opening displacements
```

```fortran
c
      do j = 1, 4
        ds1(j)=coord_l(1,j+4)-coord_l(1,j)
        ds2(j)=coord_l(2,j+4)-coord_l(2,j)
        dn(j) =coord_l(3,j+4)-coord_l(3,j)
      end do
c
c Determine the values of the shape function at each Gauss Point
c
      call k_shape_fun(i,sf)
c
      call k_vector_zero(delu_loc_gp,mcrd)
c
c Determine shear and normal opening displacements at Gauss points
c
      do j = 1, 4
        delu_loc_gp(1)=delu_loc_gp(1)+ds1(j)*sf(j)
        delu_loc_gp(2)=delu_loc_gp(2)+ds2(j)*sf(j)
        delu_loc_gp(3)=delu_loc_gp(3)+ dn(j)*sf(j)
      end do
c
      opn=delu_loc_gp(3)
      opt=sqrt(delu_loc_gp(1)**2.0+delu_loc_gp(2)**2.0)
c
      if ((Svars(GP_n*(i-1.0)+1.0) .LT. opt) .AND.
   & (opt .GT. rt*delt)) then
        Svars(GP_n*(i-1.0)+1.0)=opt
      end if
      if ((Svars(GP_n*(i-1.0)+2.0) .LT. opn) .AND.
   & (opn .GT. rn*deln)) then
        Svars(GP_n*(i-1.0)+2.0)=opn
      end if
      tmax=Svars(GP_n*(i-1.0)+1.0)
      pmax=Svars(GP_n*(i-1.0)+2.0)
c
c Determine Traction vector and tangent modulus matrix
c
call k_cohesive_law(Trac,Trac_Jacob,G_fn,G_ft,deln,delt,
```

```fortran
     &alpha,beta,p_m,p_n,pmax,tmax,alphaV,betaV,alphaP,delu_loc_gp,
     &mcrd, nrhs, SVARS)
c
c Determine B matrix and its transpose
c
      call k_Bmatrix(sf,B,mcrd,ndofel)
c
      call k_matrix_transpose(B,B_t,mcrd,ndofel)
c
c Compute the stiffness matrix
c
      call k_matrix_multiply(Trac_Jacob,B,temp1,mcrd,mcrd,
     & ndofel)
      call k_matrix_multiply(B_t,temp1,stiff_l,ndofel,
     & mcrd,ndofel)
c
c Compute Global stiffness matrix
c
      call k_matrix_multiply(Transformation_M_T,stiff_l,
     & temp2,ndofel,ndofel,ndofel)
      call k_matrix_multiply(temp2,Transformation_M,stiff_g,
     & ndofel,ndofel,ndofel)
c
c Multiply Jacobian with the Global stiffness and add contribution from each Gauss Point
c
      call k_matrix_plus_scalar(amatrx,stiff_g,a_Jacob,
     & ndofel,ndofel)
c
c Compute the global residual vector
c Local_residual = B_t * Trac
c Global_residual = T' * Local_residual
c
      call k_matrix_multiply(B_t,Trac,residual_l,ndofel,
     & mcrd,nrhs)
      call k_matrix_multiply(Transformation_M_T,residual_l,
     & residual_g,ndofel,ndofel,nrhs)
c
c Multiply the Global residual by the Jacobian and add the
```

```fortran
c contribution from each point
c
      call k_matrix_plus_scalar(rhs,residual_g,a_Jacob,
     & ndofel,nrhs)
      end do
c
      return
      end
c=======================SUBROUTINES=======================

c Determine the strain-displacement (B) matrix
c
      subroutine k_Bmatrix(sf,B,mcrd,ndofel)
      INCLUDE 'ABA_PARAM.INC'
      dimension sf(4),B(mcrd,ndofel)
      B(1,1) =  sf(1)
      B(1,4) =  sf(2)
      B(1,7) =  sf(3)
      B(1,10)=  sf(4)
      B(1,13)= -sf(1)
      B(1,16)= -sf(2)
      B(1,19)= -sf(3)
      B(1,22)= -sf(4)
      B(2,2) =  sf(1)
      B(2,5) =  sf(2)
      B(2,8) =  sf(3)
      B(2,11)=  sf(4)
      B(2,14)= -sf(1)
      B(2,17)= -sf(2)
      B(2,20)= -sf(3)
      B(2,23)= -sf(4)
      B(3,3) =  sf(1)
      B(3,6) =  sf(2)
      B(3,9) =  sf(3)
      B(3,12)=  sf(4)
      B(3,15)= -sf(1)
      B(3,18)= -sf(2)
      B(3,21)= -sf(3)
```

```fortran
      B(3,24)= -sf(4)
c
      return
      end
c================================================================

      subroutine k_cohesive_law(T,T_d,G_fn,G_ft,deln,delt,
     & alpha,beta,p_m,p_n,pmax,tmax,alphaV,betaV,alphaP,delu,
     & mcrd, nrhs, SVARS)
      INCLUDE 'ABA_PARAM.INC'
      dimension T(mcrd,nrhs),T_d(mcrd,mcrd),delu(mcrd),SVARS(38)
       DOUBLE PRECISION G_fn, G_ft, f_tn, f_tt, alpha, beta,
     & alphaV, betaV, alphaP, p_m, p_n, deln, delt, tmax, pmax,
     & popn, popt, gam_n, gam_t, Tn, Tt, Dnn, Dnt, Dtt, T_d, T,
     & delu
c
      popn=delu(3)
      popt=sqrt(delu(1)**2.0+delu(2)**2.0)
c
      call k_Mac(pM1,G_fn,G_ft)
      call k_Mac(pM2,G_ft,G_fn)
c
      if (G_fn .NE. G_ft) then
        gam_n=(-G_fn)**(pM1/(G_fn-G_ft))*(alpha/p_m)**p_m
        gam_t=(-G_ft)**(pM2/(G_ft-G_fn))*(beta/p_n)**p_n
      elseif (G_fn .EQ. G_ft) then
        gam_n=-G_fn*(alpha/p_m)**p_m
        gam_t=(beta/p_n)**p_n
      end if
c
c Pre-calculation of the normal cohesive traction Tn
c
      if (popn .LT. 0.0) then
        popn=0.0
      elseif ((popn .GE. deln) .OR. (popt .GE. delt)) then
        Tn = 0.0
      elseif (popn .GE. pmax) then
        Tn=(gam_n/deln)*(p_m*(1.0-(popn/deln))**alpha*((p_m/alpha)
```

67

```fortran
     & +(popn/deln))**(p_m-1.0)-alpha*((1.0-(popn/deln))**(alpha-1.0))
     & *((p_m/alpha)+(popn/deln))**p_m)*(gam_t*(1.0-(popt/delt))
     & **beta*((p_n/beta)+(popt/delt))**p_n+pM2)
      else
        Tn=(gam_n/deln)*(p_m*(1.0-(pmax/deln))**alpha*((p_m/alpha)
     & +(pmax/deln))**(p_m-1.0)-alpha*((1.0-(pmax/deln))**(alpha-1.0))
     & *((p_m/alpha)+(pmax/deln))**p_m)*(gam_t*(1.0-(popt/delt))
     & **beta*((p_n/beta)+(popt/delt))**p_n+pM2)*popn/pmax
      end if
c
c
c Algortihm 1
c
c Normal Cohesive Interaction
c (1) Contact
      if (delu(3) .LT. 0.0) then
        Dnn = -(gam_n/(deln**2))*(p_m/alpha)**(p_m-1.0)*(alpha+p_m)*
     & (gam_t*(p_n/beta)**p_n + pM2)
        Dnt = 0.0
        Tn = Dnn * delu(3)
       else if ((popn .LT. deln) .AND. (popt .LT. delt)
     & .AND. (Tn .GE. -1.0E-5)) then
        Tn = Tn
c (2) Softening Condition
        if (popn .GE. pmax) then
        Dnn=(gam_n/(deln**2.0))*((p_m**2.0-p_m)*((1.0-(popn/deln))
     & **alpha)*((p_m/alpha)+(popn/deln))**(p_m-2.0)+(alpha**2.0-alpha)
     & *((1.0-(popn/deln))**(alpha-2.0))*((p_m/alpha)+(popn/deln))**p_m
     & -2.0*alpha*p_m*((1.0-(popn/deln))**(alpha-1.0))*((p_m/alpha)
     & +(popn/deln))**(p_m-1.0))*(gam_t*(1.0-(popt/delt))**beta
     &*((p_n/beta)+(popt/delt))**p_n+pM2)
        Dnt=(gam_n*gam_t/(deln*delt))*(p_m*((1.0-(popn/deln))**alpha)
     & *((p_m/alpha)+(popn/deln))**(p_m-1.0)-alpha*((1.0-(popn/deln))
     & **(alpha-1.0))*((p_m/alpha)+(popn/deln))**p_m)*(p_n
     & *((1.0-(popt/delt))**beta)*(((p_n/beta)+(popt/delt))**(p_n-1.0))
     & -beta*((1.0-(popt/delt))**(beta-1.0))*((p_n/beta)+(popt/delt))
     & **p_n)
c (3) Unloading/reloading condition
```

```fortran
      else
      Dnn=(gam_n/deln)*(p_m*(1.0-(pmax/deln))**alpha*((p_m/alpha)
     & +(pmax/deln))**(p_m-1.0)-alpha*((1.0-(pmax/deln))**(alpha-1.0))
     & *((p_m/alpha)+(pmax/deln))**p_m)*(gam_t*(1.0-(popt/delt))
     & **beta*((p_n/beta)+(popt/delt))**p_n+pM2)/pmax
      Dnt=(gam_n*gam_t/(deln*delt))*(p_m*((1.0-(pmax/deln))**alpha)
     & *((p_m/alpha)+(pmax/deln))**(p_m-1.0)-alpha*((1.0-(pmax/deln))
     & **(alpha-1.0))*((p_m/alpha)+(pmax/deln))**p_m)*(p_n
     & *((1.0-(popt/delt))**beta)*(((p_n/beta)+(popt/delt))**(p_n-1.0))
     & -beta*((1.0-(popt/delt))**(beta-1.0))*((p_n/beta)+(popt/delt))
     & **p_n)*popn/pmax
      end if
c (4) Complete Failure
       else
         Tn = 0.0
         Dnn = 0.0
         Dnt = 0.0
       end if
c
c===============================================================
      subroutine k_local_coordinates(gpt,co_de,R,coord_l,Transformation_M,
     & Transformation_M_T,a_Jacob,aJacob_M,coords,u,ndofel,nnode,
     & mcrd, SVARS)
       INCLUDE 'ABA_PARAM.INC'
       dimension R(mcrd,mcrd),coord_l(mcrd,nnode),aJacob_M(2,3),
     & Transformation_M(ndofel,ndofel),coords(mcrd,nnode),
     & Transformation_M_T(ndofel,ndofel),u(ndofel),
     & co_de(mcrd,nnode), co_de_m(3,4),SFD(2,4),SVARS(38)
        DOUBLE PRECISION x1, x2, x3, x4, y1, y2, y3, y4, y5, y6, z1, z2,
     & z3, z4
c
      call k_matrix_zero(co_de_m,3,4)
c
      do i = 1, 3
        co_de_m(i,1)=(co_de(i,1)+co_de(i,5))*0.5
        co_de_m(i,2)=(co_de(i,2)+co_de(i,6))*0.5
        co_de_m(i,3)=(co_de(i,3)+co_de(i,7))*0.5
        co_de_m(i,4)=(co_de(i,4)+co_de(i,8))*0.5
```

```
      end do
c
      x1=co_de_m(1,1)
      x2=co_de_m(1,2)
      x3=co_de_m(1,3)
      x4=co_de_m(1,4)
c
      y1=co_de_m(2,1)
      y2=co_de_m(2,2)
      y3=co_de_m(2,3)
      y4=co_de_m(2,4)
c
      z1=co_de_m(3,1)
      z2=co_de_m(3,2)
      z3=co_de_m(3,3)
      z4=co_de_m(3,4)
c
      if (gpt .eq. 1) then
        c_r=-sqrt(1.0d0/3.0d0)
        c_s=-sqrt(1.0d0/3.0d0)
      elseif (gpt .eq. 2) then
        c_r= sqrt(1.0d0/3.0d0)
        c_s=-sqrt(1.0d0/3.0d0)
      elseif (gpt .eq. 3) then
        c_r= sqrt(1.0d0/3.0d0)
        c_s= sqrt(1.0d0/3.0d0)
      elseif (gpt .eq. 4) then
        c_r=-sqrt(1.0d0/3.0d0)
        c_s= sqrt(1.0d0/3.0d0)
      end if
c
      SFD(1,1) =-0.25*(1-c_s)
      SFD(1,2) = 0.25*(1-c_s)
      SFD(1,3) = 0.25*(1+c_s)
      SFD(1,4) =-0.25*(1+c_s)
      SFD(2,1) =-0.25*(1-c_r)
      SFD(2,2) =-0.25*(1+c_r)
      SFD(2,3) = 0.25*(1+c_r)
```

```fortran
      SFD(2,4) = 0.25*(1-c_r)
c
      do i = 1,2
        do j = 1,3
          do k =1, 4
            aJacob_M(i,j) = aJacob_M(i,j) + SFD(i,k)*co_de_m(j,k)
          end do
        end do
      end do
c
      dum1 = aJacob_M(1,2)*aJacob_M(2,3) - aJacob_M(1,3)*aJacob_M(2,2)
      dum2 = aJacob_M(1,3)*aJacob_M(2,1) - aJacob_M(1,1)*aJacob_M(2,3)
      dum3 = aJacob_M(1,1)*aJacob_M(2,2) - aJacob_M(1,2)*aJacob_M(2,1)
c
      a_Jacob = sqrt(dum1**2 + dum2**2 + dum3**2)
c
      R(3,1) = dum1/a_Jacob
      R(3,2) = dum2/a_Jacob
      R(3,3) = dum3/a_Jacob
c

aLen=sqrt(aJacob_M(1,1)**2.0d00+aJacob_M(1,2)**2.0d00+aJacob_M(1,3)**2.0d00)
      R(1,1)=aJacob_M(1,1)/aLen
      R(1,2)=aJacob_M(1,2)/aLen
      R(1,3)=aJacob_M(1,3)/aLen
c
      aLen2a=R(3,2)*R(1,3)-R(3,3)*R(1,2)
      aLen2b=R(3,3)*R(1,1)-R(3,1)*R(1,3)
      aLen2c=R(3,1)*R(1,2)-R(3,2)*R(1,1)
c
      aLen2 = sqrt(aLen2a**2.0d00 + aLen2b**2.0d00 + aLen2c**2.0d00)
c
      R(2,1) = aLen2a/aLen2
      R(2,2) = aLen2b/aLen2
      R(2,3) = aLen2c/aLen2
c
      a_J11 = (-0.25*(1-c_s))*x1 + (0.25*(1-c_s))*x2 + (0.25*(1+c_s))*x3 +
     & (-0.25*(1+c_s))*x4
```

```
      a_J12 = (-0.25*(1-c_s))*z1 + (0.25*(1-c_s))*z2 + (0.25*(1+c_s))*z3 +
     & (-0.25*(1+c_s))*z4
      a_J21 = (-0.25*(1-c_r))*x1 + (-0.25*(1+c_r))*x2 + (0.25*(1+c_r))*x3 +
     & (0.25*(1-c_r))*x4
      a_J22 = (-0.25*(1-c_r))*z1 + (-0.25*(1+c_r))*z2 + (0.25*(1+c_r))*z3 +
     & (0.25*(1-c_r))*z4
c
      b_J11 = (-0.25*(1-c_s))*x1 + (0.25*(1-c_s))*x2 + (0.25*(1+c_s))*x3 +
     & (-0.25*(1+c_s))*x4
      b_J12 = (-0.25*(1-c_s))*y1 + (0.25*(1-c_s))*y2 + (0.25*(1+c_s))*y3 +
     & (-0.25*(1+c_s))*y4
      b_J21 = (-0.25*(1-c_r))*x1 + (-0.25*(1+c_r))*x2 + (0.25*(1+c_r))*x3 +
     & (0.25*(1-c_r))*x4
      b_J22 = (-0.25*(1-c_r))*y1 + (-0.25*(1+c_r))*y2 + (0.25*(1+c_r))*y3 +
     & (0.25*(1-c_r))*y4
c
      c_J11 = (-0.25*(1-c_s))*y1 + (0.25*(1-c_s))*y2 + (0.25*(1+c_s))*y3 +
     & (-0.25*(1+c_s))*y4
      c_J12 = (-0.25*(1-c_s))*z1 + (0.25*(1-c_s))*z2 + (0.25*(1+c_s))*z3 +
     & (-0.25*(1+c_s))*z4
      c_J21 = (-0.25*(1-c_r))*y1 + (-0.25*(1+c_r))*y2 + (0.25*(1+c_r))*y3 +
     & (0.25*(1-c_r))*y4
      c_J22 = (-0.25*(1-c_r))*z1 + (-0.25*(1+c_r))*z2 + (0.25*(1+c_r))*z3 +
     & (0.25*(1-c_r))*z4
c
c
      a_Jacob1 = (a_J11*a_J22 - a_J12*a_J21)
      a_Jacob2 = (b_J11*b_J22 - b_J12*b_J21)
      a_Jacob3 = (c_J11*c_J22 - c_J12*c_J21)
c
      a_Jacob = sqrt(a_Jacob1**2.0d00 + a_Jacob2**2.0d00 + a_Jacob3**2.0d00)
c
c===================================================================
=
      num=nnode
c
      do i = 1, num
        dum=3.0*(i-1.0)
```

```fortran
      Transformation_M(dum+1,dum+1)=R(1,1)
      Transformation_M(dum+1,dum+2)=R(1,2)
      Transformation_M(dum+1,dum+3)=R(1,3)
      Transformation_M(dum+2,dum+1)=R(2,1)
      Transformation_M(dum+2,dum+2)=R(2,2)
      Transformation_M(dum+2,dum+3)=R(2,3)
      Transformation_M(dum+3,dum+1)=R(3,1)
      Transformation_M(dum+3,dum+2)=R(3,2)
      Transformation_M(dum+3,dum+3)=R(3,3)
      end do
c
      call k_matrix_transpose(Transformation_M,Transformation_M_T,
     $ ndofel,ndofel)
c
c
      do i = 1, nnode
        coord_l(1,i)=(R(1,1)*co_de(1,i)+R(1,2)*co_de(2,i)
     & +R(1,3)*co_de(3,i))
        coord_l(2,i)=(R(2,1)*co_de(1,i)+R(2,2)*co_de(2,i)
     & +R(2,3)*co_de(3,i))
        coord_l(3,i)=(R(3,1)*co_de(1,i)+R(3,2)*co_de(2,i)
     & +R(3,3)*co_de(3,i))
      end do
c
      return
      end
c==================================================================
      subroutine k_shape_fun(i,sf)
      INCLUDE 'ABA_PARAM.INC'
      dimension sf(4), GP_coord(2)
c
      if (i .eq. 1) then
        GP_coord(1)=-sqrt(1.0d0/3.0d0)
        GP_coord(2)=-sqrt(1.0d0/3.0d0)
      elseif (i .eq. 2) then
        GP_coord(1)= sqrt(1.0d0/3.0d0)
        GP_coord(2)=-sqrt(1.0d0/3.0d0)
```

```fortran
      elseif (i .eq. 3) then
        GP_coord(1)= sqrt(1.0d0/3.0d0)
        GP_coord(2)= sqrt(1.0d0/3.0d0)
      elseif (i .eq. 4) then
        GP_coord(1)=-sqrt(1.0d0/3.0d0)
        GP_coord(2)= sqrt(1.0d0/3.0d0)
      end if
c
      sf(1)=(1-GP_coord(1))*(1-GP_coord(2))*0.25
      sf(2)=(1+GP_coord(1))*(1-GP_coord(2))*0.25
      sf(3)=(1+GP_coord(1))*(1+GP_coord(2))*0.25
      sf(4)=(1-GP_coord(1))*(1+GP_coord(2))*0.25
c
      return
      end
c=================================================================
      subroutine k_matrix_multiply(A,B,C,l,n,m)
      INCLUDE 'ABA_PARAM.INC'
      dimension A(l,n),B(n,m),C(l,m)
c
      call k_matrix_zero(C,l,m)
c
      do i = 1, l
        do j = 1, m
          do k = 1, n
            C(i,j)=C(i,j)+A(i,k)*B(k,j)
          end do
        end do
      end do
c
      return
      end
c=================================================================
      subroutine k_matrix_plus_scalar(A,B,c,n,m)
      INCLUDE 'ABA_PARAM.INC'
      dimension A(n,m),B(n,m)
```

```
c
      do i = 1, n
        do j = 1, m
          A(i,j)=A(i,j)+c*B(i,j)
        end do
      end do
c
      return
      end
c================================================================
=
      subroutine k_matrix_transpose(A,B,n,m)
      INCLUDE 'ABA_PARAM.INC'
      dimension A(n,m),B(m,n)
c
      do i = 1, n
        do j = 1, m
          B(j,i)=A(i,j)
        end do
      end do
c
      return
      end
c================================================================
=
      subroutine k_matrix_zero(A,n,m)
      INCLUDE 'ABA_PARAM.INC'
      dimension A(n,m)
c
      do i = 1, n
        do j = 1, m
          A(i,j)=0.d0
        end do
      end do
c
      return
      end
```

```fortran
c==================================================================
=
      subroutine k_vector_zero(A,n)
      INCLUDE 'ABA_PARAM.INC'
      dimension A(n)
c
      do i = 1, n
        A(i)=0.d0
      end do
c
      return
      end
c==================================================================
=
      subroutine k_Mac(pM,a,b)
      INCLUDE 'ABA_PARAM.INC'
c
      if ((a-b) .GE. 0.0) then
        pM=a-b
      elseif ((a-b) .LT. 0.0) then
        pM=0.d0
      end if
c
      return
      end
c===========================END===========================
```