Enrupture Analysis Framework Guide

March 8, 2017

Contents

1 Overview

Enrupture Analysis Framework (EAF) is an interactive tool for exploration, visualization and comparison of Policrack databases, built upon Jupyter Notebook.

It aims to show the capabilities and limitations of FEM and XFEM fracture analyses of elliptic cracks modeled with different strategies and element types.

EAF has been used to review and analyze a dataset at the scale of tens of thousands of records effortlessly.

2 Concepts

2.1 Directory structure

The directory structure is shown in Figure 1. The **db** directory is expected to be the Policrack database repository, on which EAF will operate.

The **enrupture** directory contains the EAF files. The **shardlib** directory contains the **shardlib** library, which provides the functionality of EAF. Documentation is in the **docs** directory. Session files are stored in **sessiondata**.

```
project directory
 ├─ db ..................................... Policrack database repository
 │   ├─ database 1
 │   ├─ ...
 │   └─ database n
 └─ enrupture ........................................... EAF directory
     ├─ shardlib
     ├─ docs
     └─ sessiondata
```

Figure 1: Directory structure

2.2 Simulation Id – simId

For each database record EAF creates a simId, which uses to internally identify that record. Working with EAF you may never see the actual simId string. The simId is guaranteed to be unique and EAF utilizes it to retrieve record data efficiently. The format of a simId is:

`database_name` + `$` + `record_key_in_database`.

2.2.1 Simulation Id Status

EAF distinguishes two types of simIds – successful and failed. All analyses are performed on simIds with successful status. To exclude a simId from further analysis its status is changed to failed. Failed simIds are counted. A simId is considered successful if it has a 'successfulAnalysis' flag set to True and is not in the session file for failed simIds.

2.3 Database Management

A Policrack repository may contain one or more databases. EAF simplifies database operations (query, retrieval) by using an single index of all records of all databases. EAF only supports read operations on the Policrack databases.

2.3.1 Index Tree

The index is a tree – hierarchical data structure, used to classify a given set of simIds based on crack ratio, analysis type, model type and elements. Each node of the tree keeps track of simIds corresponding to records of successful and failed simulations independently. EAF creates several indexes, the first index is created from the complete set of databases and records, others are created from a subsets of records. All of EAF analyses depend on an index.

2.3.2 Queries

EAF supports two types of queries – index search and table scans.

Index search is the selection of a node from an index tree. The query is based on organization of simIds in the index. EAF does not need to read the database to perform the search and is much faster than a table scan. An index search is a way to quickly filter and focus on particular type of simulations.

Table scan queries are performed on a set of simIds. EAF needs to retrieve each record for the given set of simIds to determine the whether the simId meets the query criteria.

2.3.3 Retrieval

EAF retrieves a record from its simId. The database management system recreates the path to the database and the record key from the simId.

2.4 Session Persistence

EAF does not write to the Policrack databases. However, it does provide a mechanism to store simIds between sessions into session files. Session files are stored in the **sessiondata** directory Figure 1. The session files are two Python shelves for successful and failed simIds. Unless new session files are created, the last session files are loaded either as indexes updates or added to a queue.

When a simId is designated as failed, all indexes are updated and the simId is not taken further into consideration.

To remove a failed simId from a session file, either manually edit the shelf or initialize new session files. When EAF creates a new pair of session files, it backs up the old ones.

2.5 Queues

A queue is used to add, store, retrieve, remove and print information about the stored entities throughout an EAF session. It allows for convenient manipulation of the entities, without dealing with them directly. Queues are two types, based on the type of entities:

- CompQueue (CQ) – used to store simIds. CQs are also used to load simIds from the successful session file.
- LeavesQueue (LQ) – used to store index tree nodes.

2.6 Parameters and Values

2.6.1 Input Parameters

A set of input parameters completely defines a Policrack model used to calculate the record data. Input parameters are:

container size - diameter and height of the cylinder containing the crack

crack ratio - the ratio of the ellipse axes

initial conditions - σ - stress applied to the container, ω - angle of rotation of the crack plane along its normal, γ - angle of rotation of the crack plane along one of ellipse axes.

analysis type - either FEM or XFEM

model type - determines the modeling strategy

element type - determines the used elements

2.6.2 Calculated and Analytical Data

Record data obtained from a FEM analysis is referred as calculated data. The corresponding analytical data is obtained from analytical solutions using the same input parameters. Analytical data is not stored in the Policrack databases and is recalculated on demand.

2.6.3 Estimates

EAF derives record data estimates either from the calculated Stress Intensity Factors (SIFs) values only or from calculated and analytical values.

areaDiff - A (Normalized difference between integrals of the analytical and analysis along the crack front) - areaDiff is the normalized difference between the area of the calculated data and the analytical solution. It estimates whether the simulation underestimates ($A < 0$), or overestimates ($A > 0$) the analytical solution.

$$A = \frac{\sum_i^{p-1} \left[(\alpha_{i+1} - \alpha_i) (F_{i+1} + F_i) \right] - \sum_i^{p-1} \left[(\alpha_{i+1} - \alpha_i) (S_{i+1} + S_i) \right]}{\sum_i^{p-1} \left[(\alpha_{i+1} - \alpha_i) (S_{i+1} + S_i) \right]},$$

where α_i is the angle of the i^{th} node of the crack front; S_i is the analytical SIF value of the i^{th} node of the crack front; F_i is the analysis SIF value of the i^{th} node of the crack front; p the number of nodes along the crack front.

avgNormError - E (Normalized average error along the crack front)

$$E = \frac{A}{|A|} . \frac{\sum_i^p \left| \frac{S_i - F_i}{\max(S)} \right|}{p},$$

where A is the areaDiff; S_i is the analytical SIF value of the i^{th} node of the crack front; F_i is the analysis SIF value of the i^{th} node of the crack front; p the number of nodes along the crack front.

dotProd - D (Normalized error difference between analytical and analysis dot products)

$$D = \frac{\sum_i^p F_i^2}{\sum_i^p S_i^2} - 1,$$

where S_i is the analytical SIF value of the i^{th} node of the crack front; F_i is the analysis SIF value of the i^{th} node of the crack front; p the number of nodes along the crack front.

maxNormError - M (Maximum magnitude of the normalized error along the crack front)

$$M = \frac{\max(|F_i - S_i|)}{\max(|S_i|)},$$

where S are the analytical SIF values along the crack front; F are the analysis SIF values along the crack front;

rmsd - R (Root mean square deviation)

$$R = \sqrt{\frac{\sum_i^p (S_i - F_i)^2}{p}},$$

where S_i is the analytical SIF value of the i^{th} node of the crack front; F_i is the analysis SIF value of the i^{th} node of the crack front; p the number of nodes along the crack front.

2.6.4 Errors

Errors are calculated as the difference between the calculated values for the SIFs along the crack front and the analytical values for the same points. Errors can be normalized or non-normalized (difference).

2.7 Optimal Simulation – optSim

An optimal simulation is a simId, for which best satisfies a criterion. A criterion can be the minimization of an error value obtained for a given SIF.

The optSim is used as a reference for comparison. It is the simId from a set, which is closest to the analytical solution, based on the given criteria.

2.8 Analyses

2.8.1 Container Size Convergent

Container Size Convergence analysis (CSC) in EAF is performed by the means of contour plots for a set of simIds. Subject to the CSC analysis is a set of simIds with equal crack ratio. The set can be any valid combination of analysis, model and element types.

The $x-$ and $y-$ axes represent the container diameter d and height h. The available configurations of container sizes form a grid of configuration points, represented by black dots. The plotted variable is a selected estimate for a SIF, calculated at each configuration point. The estimate at each point is calculated from an optSim selected for that point.

The CSC analysis is performed in three stages – preview, selection of container size and verification. The main difference between the preview and verification stages is how the optSims are selected.

For the the preview stage the criteria parameter and SIF for the optSim selection is tied to the estimate (criteria parameter) and SIF for the plot. In other words, this selection of optSims may underestimate the necessary container size for convergence.

In the verification stage the criteria parameter and SIF are predefined and independent of the contour plot.

The plots should give clear indication of the point at which convergence is reached unless the noise in the data is high.

2.8.2 Comparative Analyses

The Comparative Analyses (CA) are used for juxtaposing record data in a plot. Comparative analyses are two types single simId and multiple simId, depending on how data is derived and presented.

Single simId CA – SSCA

SSCA is performed with box plots, where each box represents one simId from a given set of simIds. Each successful simId in a single simId CA plot is assigned a number, labeled as `Selection ID Number`, which can be entered as an argument in a function in a subsequent notebook cell. This allows for setting a simId status to failed or to be added to a queue. Single simId CA are XCPCompAnalysis, XSCompAnalysis and FEMCompAnalysisis and each is used for simIds of a particular model type.

XCPCompAnalysis is used for XFEM crackPartition model type. The left column of the box plot displays the optSims for the subsets of simIds with equal crackEdge value. The right column displays all the simIds from a subset with a designated crackEdge.

XSCompAnalysis is used for XFEM simple model type. Each column of subplots displays a subset of simIds with a designated element type.

FEMCompAnalysis is used for all FEM model types. Each column of subplots displays a combination of a designated FEM model and element type.

Multiple simId CA – MSCA
In MSCA, is performed on aggregated data from a set of simIds. MSCA is performed in the section *Comparison of the Model Types* of EAF. The following plots are provided in the section:

BoxCompPlot is a box plot similar to SSCA, however, boxes represent sets of aggregated data.

HistCompPlot is a histogram of an estimate for each set of aggregated data.

CorrCompPlot shows how an estimate correlates to the value of the analytical solution of the data point. Each point in the plot represents an estimate of a data point plotted versus the analytical value of the data point. The black line is a reference for perfect correlation.

RangeCompPlot shows the range of variability of the calculated data for a set of simIds.

BoundsCompPlot determines and visualizes an upper and lower bound within which a percentage of the calculated data points fall in. Let:

- u and l are upper and lower target percent,
- F is a calculated SIF value,
- σ_u and σ_l are the upper and lower bounds for the applied stress σ,
- S_u is an analytical value for the SIF at the same point of the crack estimated with σ_u and S_l is an analytical value for the SIF at the same point of the crack estimated with σ_l.

Then the values of σ_u is such that $|F| < |S_u|$ is true for u percent of the number of calculated values.

The value of σ_l is such that $|F| < |S_l|$ is true for l percent of the number of calculated values.

3 Analysis Workflow

The analysis process with EAF can be broken down to the following generalized activities:

initialization – identification of the databases and index creation.

selection – an entity (simId, a set of simIds or an index (tree) node).

visualization – creation of a plot.

session persistence – saving one or more simIds to the session files.

Working with EAF you may perform the activities sequentially in the order they appear. Or, in a different order, or depending on your needs, change the parameters of the functions of an activity and perform it again.

There are some limitations, like the initialization has to be performed always first and visualization has to follow selection.

4 EAF structure

1. Initialization
    (a) New Session
    Set the argument to True to backup current session files and create new ones.
    (b) Number and Distribution of the Simulations
        i. All Simulations
        Creates a tree from simIds of all simulations. Updates the tree with the failed session file. Prints and plots tree stats.
        ii. Simulations with Equal Initial Conditions and Material
        For specified initial conditions and material, filters the simIds from 1(b)i, creates a new tree, prints and plots stats.
        iii. Assignment of Crack Ratio
        Specify a valid value for crack ratio. A valid value is such that there is at least one simId with that crack ratio in 1(b)ii. All subsequent analyses in the EAF are based on simIds with that crack ratio.
        iv. Simulations with Equal Crack Ratio
        Filters simIds from 1(b)ii that have the specified crack ratio, creates a tree, prints and plots stats.
    (c) Container Size Convergence
        i. Contour Plots for CSC - Preview Stage
        Select SIFs and a valid combination of analysis, model and element types as a subject of the analysis. CSC analysis is based on a subset of simIds from 1(b)iv with the specified analysis, model and element types.
        ii. Optimal Container Size
        Specify valid cylinder diameter and height. Optimal diameter and height are those values that minimize the effects of the finite container. For diameter and height to be valid, the set of simIds in 1(b)iv should contain at least one simId with that container size.
        iii. Contour Plots for CSC - Verification Stage
        Select criteria parameter and SIF for optSim, SIFs. Estimator values for the plots are calculated from optSims selected from the specified criteria parameter and SIF.
    (d) Number and Distribution of Simulations with Optimal Size Container
    Filters simIds from 1(b)iv with the container size of 1(c)ii, creates a tree, prints and plots stats. All subsequent analyses are performed on simIds from the nodes of this tree, unless stated otherwise.

2. Analysis of the Model Types
    (a) XFEM crackPartition
        i. Number and Distribution of XFEM CP Simulations

A. Stats for all XFEM CP Simulations with the Specified Crack Ratio
Prints stats of the XFEM CP simulations of 1(b)iv

B. Successful Simulations vs Seed Parmaters Plots
Creates plots of the successful and failed XFEM CP simulations vs container size vs allEdges for the each value of the crackEdge parameter.

C. Stats for all XFEM CP Simulations with the Specified Crack Ratio and Optimal Container Size
Prints stats of the XFEM CP simulations of 1d

ii. Initialization
Create CompQueue, add to the CQ relevant simIds from the session persistence file successful, select criteria and SIF for optSim, select SIFs for consideration in 2(a)iii.

iii. Analysis Step

A. Box Plot
Specify a valid crackEdge value for the box plot. A valid crackEdge value is such that there is at least one XFEM CP simulation with that value of crackEdge in 1d.

B. Assignment of Failed simIds
Should a simulation is deemed failed, select a number corresponding to the box in the box plot in 2(a)iiiA to assign a simulation as failed. The simId is saved to the failed session persistence file, trees are updated, and the simId is removed from the CQ (if it has been added). Repeat as many times as necessary.

C. Add simIds to CQ
Select a number corresponding to a box in the box plot in 2(a)iiiA to add to the CQ. Repeat as many times as necessary.

D. SIFs Plots of simIds in the CQ
Prints stats of the simIds in CQ. Creates SIFs plots with analytical solution for every simId in the CQ

E. Remove a simId from the CQ
Should a simulation needs to be removed from the CQ, select a number corresponding to the simId in the CQ. Repeat as many times as necessary.

F. Errors and SIFs Plots for a Selected simId
Plots errors and SIFs for a selected number of a simId in the CQ.

G. Save a Selected simId to the Session successful File
Saves to the session persistence successful file the simId from the CQ with number selected in 2(a)iiiF.

(b) XFEM simple

i. Number and Distribution of XFEM simple Simulations

A. Stats for all XFEM simple Simulations with the Specified Crack Ratio
Prints stats of the XFEM simple simulations of 1(b)iv

B. Successful Simulations vs Seed Parameters Plots
Creates plots of the successful and failed XFEM simple simulations vs container size vs allEdges for the each type of elements.

C. Stats for all XFEM simple Simulations with the Specified Crack Ratio and Optimal Container Size
Prints stats of the XFEM simple simulations of 1d

ii. Initialization
Create CompQueue, add to the CQ relevant simIds from the session persistence file successful, select criteria and SIF for optSim, select SIFs for consideration in 2(b)iii.

iii. Analysis Step

A. Box Plot
Specify one or two valid element types for the box plot. A valid element type is such that there is at least one XFEM simple simulation with that element type 1d.

B. Assignment of Failed simIds
Should a simulation is deemed failed, select a number corresponding to the box in the boxplot in 2(b)iiiA to assign a simulation as failed. The simId is saved to the failed session persistence file, trees are updated, and the simId is removed from the CQ (if it has been added). Repeat as many times as necessary.

C. Add simIds to CQ
Select a number corresponding to a box in the box plot in 2(b)iiiA to add to the CQ. Repeat as many times as necessary.

D. SIFs Plots of simIds in CQ
Prints stats of the simIds in CQ. Creates SIFs plots with analytical solution for every simId in the CQ

E. Remove a simId from the CQ
Should a simulation needs to be removed from the CQ, select a number corresponding to the simId in the CQ. Repeat as many times as necessary.

F. Errors and SIFs Plots for a Selected simId
Plots errors and SIFs for a selected number of a simId in the CQ.

G. Save a Selected simId to the Session successful File
Saves to the session persistence successful file the simId from the CQ with number selected in 2(b)iiiF.

(c) FEM

i. Container size comparison for XFEM and FEM analyses
Plots the top and side views of container sizes for XFEM simulations and FEM scale and FEM elliptic model types. Due to the mesh transformations, the circular cross-section is elongated to a different degree for the FEM model types.

ii. Number and Distribution of FEM Simulations

A. Stats for all FEM Simulations with the Specified Crack Ratio
Prints stats of the FEM simulations of 1(b)iv

B. Stats for all FEM Simulations with the Specified Crack Ratio and Optimal Container Size
Prints stats of the FEM simulations of 1d

iii. Initialization
Create CompQueue, add to the CQ relevant simIds from the session persistence file successful, select criteria and SIF for optSim, select SIFs for consideration in 2(c)iv.

iv. Analysis Step

A. Box Plot
Specify one or two valid combinations of FEM model and element types for the box plot. A valid combination is such that there is at least one FEM simulation of that combination 1d.

B. Assignment of Failed simIds
Should a simulation is deemed failed, select a number corresponding to the box in the box plot in 2(c)ivA to assign a simulation as failed. The simId is saved to the failed session persistence file, trees are updated, and the simId is removed from the CQ (if it has been added). Repeat as many times as necessary.

C. Add simIds to CQ
Select a number corresponding to a box in the box plot in 2(c)ivA to add to the CQ. Repeat as many times as necessary.

D. SIFs Plots of simIds in CQ
Prints stats of the simIds in CQ. Creates SIFs plots with analytical solution for every simId in the CQ

E. Remove a simId from the CQ
Should a simulation needs to be removed from the CQ, select a number corresponding to the simId in the CQ. Repeat as many times as necessary.

F. Errors and SIFs Plots for a Selected simId
Plots errors and SIFs for a selected number of a simId in the CQ.

G. Save a Selected simId to the Session successful File
Saves to the session persistence successful file the simId from the CQ with number selected in 2(c)ivF.

3. Comparison of the Model Types

(a) Initialization
Create LeavesQueue and add the leaves nodes from the tree in 1d. Specify criteria and SIF for optSim, select SIFs for consideration in 3b.

(b) Analysis Step

i. Box Plot
Specify the indexes of the leaves in LQ, to include in the plot and error type.

ii. Histogram
Create a dictionary with keys - indexes of leaves in the LQ and values - the number of bins for the histogram of the leaf. Specify error type and limits for the x-axis.

iii. Correlation Plot
Specify the indexes of the leaves in LQ, error type and y-axis limit.

iv. Range Plot
Create a dictionary wit one or two elements, keys representing the indexes of the leaves in LQ and values - the number of intervals into which to divide the data points. Specify values of the options.

v. Bounds Plot
Specify upper and lower target percentage values, tolerance and the maximum number of iterations.