experiment_interface Documentation

Release 0.0.0

Lucas Guesser Targino da Silva

Nov 04, 2019

CONTENTS

CHAPTER

ONE

INTRODUCTION

Set of tools for processing the data of mechanical testings.

Contents:

1.1 TensileTest

class mechanical_testing.**TensileTest** (*file*, *length*, *diameter*)

Process tensile testing data.

Load a tensile test data and process it in order to deliver the material properties.

> **Warning:** All values are meant to be in the SI units. Since no unit conversion is made, the input data has to be in the SI units.

originalFile

Path to the file from which the data was read.

Type str

force

Force data from the tensile test.

Type numpy.ndarray

displacement

Displacement data from the tensile test.

Type numpy.ndarray

time

Time instant data from the tensile test.

Type numpy.ndarray

length

Gage length of the specimen.

Type float

diameter

Diameter of the specimen.

Type float

area
Cross section area of the specimen. $A = \frac{\pi\ D}{4}$ being D the diameter of the specimen.

Type float

strain
Strain data of the tensile test. $\epsilon = \frac{l - l_0}{l_0} = \frac{d}{l_0}$ being l_0 the initial length.

Type numpy.ndarray

stress
Stress data of the tensile test. $\sigma = \frac{F}{A}$ being F the force and A the cross section area.

Type numpy.ndarray

realStrain
Strain for the real curve. $\epsilon_r = ln(1 + \epsilon)$.

Type numpy.ndarray

realStress
Stress for the real curve. $\sigma_r = \sigma\ (1 + \epsilon)$.

Type numpy.ndarray

proportionalityStrain, proportionalityStrength
Stress and strain values at the proportionality limit point.

Type float

yieldStrain, yieldStrength
Stress and strain values at the yield point.

Type float

ultimateStrain, ultimateStrength
Stress and strain values at the ultimate point.

Type float

strengthCoefficient, strainHardeningExponent
Those are coefficients for the Hollomon's equation during the plastic deformation. It represents the hardening behavior of the material. Hollomon's equation: $\sigma = K\ \epsilon^n$ being K the strength coefficient and n the strain hardening exponent.

Type float

elasticStrain, elasticStress
Strain and stress data when the material behaves elastically.

Type numpy.ndarray

plasticStrain, plasticStress
Strain and stress data when the material behaves plastically.

Type numpy.ndarray

neckingStrain, neckingStress
Strain and stress data when the necking starts at the material.

Type numpy.ndarray

elasticModulus
Elastic modulus value.

Type float

`resilienceModulus`
Resilience modulus value. It is the energy which the material absorbs per unit of volume during its elastic deformation.

Type float

`toughnessModulus`
Resilience modulus value. It is the energy which the material absorbs per unit of volume until its failure.

Type float

See also:

Tensile testing wikipedia page

Stress-Strain curve wikipedia page

Notes

Table 1: Title

Symbol	Description	Definition
$[F]$	force	input
$[d]$	displacement	input
$[t]$	time	input
l_0	specimen length	input
D	specimen diameter	input
A	specimen cross section area	$A = \frac{\pi D^2}{4}$
$[\epsilon]$	strain	$\epsilon = \frac{l - l_0}{l_0} = \frac{d}{l_0}$
$[\sigma]$	stress	$\sigma = \frac{F}{A}$
$[\epsilon_r]$	real strain	$\epsilon_r = ln(1 + \epsilon)$
$[\sigma_r]$	real stress	$\sigma_r = \sigma \ (1 + \epsilon)$
$\epsilon_{pr}, \sigma_{pr}$	proportionality strain and strength	algorithm defined
ϵ_y, σ_y	yield strain and strength	algorithm defined
ϵ_u, σ_u	ultimate strain and strength	algorithm defined
K	strength coefficient	algorithm defined
n	strain hardening exponent	algorithm defined
$[\epsilon_e]$	elastic strain	$[\epsilon][\epsilon < \epsilon_y]$
$[\sigma_e]$	elastic stress	$[\sigma][\epsilon < \epsilon_y]$
$[\epsilon_p]$	plastic strain	$[\epsilon][\epsilon_y < \epsilon < \epsilon_u]$
$[\sigma_p]$	plastic stress	$[\sigma][\epsilon_y < \epsilon < \epsilon_u]$
$[\epsilon_n]$	necking strain	$[\epsilon][\epsilon_u < \epsilon]$
$[\sigma_n]$	necking stress	$[\sigma][\epsilon_u < \epsilon]$
E	elastic modulus	$\sigma = E \ \epsilon$, curve fit
U_r	resilience modulus	$\int_{[\epsilon_e]} \sigma \ d\epsilon$
U_t	toughness modulus	$\int_{[\epsilon]} \sigma \ d\epsilon$

Auto-find proportionality limit and elastic modulus:

```
foreach l in range(10, len(strain)):
        fit a one-degree polynomial to the data
        store the linear coefficient
        store the curve fit residual
select the proportionality limit point as the one with the smallest residual
select the elastic modulus as the linear coefficient of the polynomial
```

Ultimate point:

```
Select the ultimate point as the one
with the maximum stress
```

Yield point:

```
select the yield point as the intersection of the curves:
        ([strain], [stress])
        ([strain], elasticModulus * ([strain]-0.002))
if the point has strain larger than the ultimate point:
        select the yield point as equals to the
        proportionality limit point
```

Hardening, strength coefficient and strain hardening exponent:

```
Curve fit (Hollomon's equation):
        f = K * strain**n
        x = [plastic strain]
        y = [plastic stress]
```

__init__(*file*, *length*, *diameter*)
Process tensile data.

Parameters

- **file** (*str*) – Path to file containing the data. The data from the file is not checked in any way. The file must be in the comma-separated-value format.
- **length** (*float*) – Length l_0 of the specimen in meters.
- **diameter** (*float*) – Diameter D of the specimen in meters.

Examples

```
>>> import mechanical_testing as mect
>>> tensile = mect.TensileTest(
            file     = './test/data/tensile/tensile_steel_1045.csv,
            length   = 75.00E-3,
            diameter = 10.00E-3,
)
>>> tensile.yieldStrength
7.6522E+8
```

offsetYieldPoint(*offset*)
Yield point defined by the input offset

Parameters **offset** (*float*) – Offset value. For the common yield point used in engineering, use *offset = 0.002 = 0.2%*.

Returns **(strain, stress)** ((*float, float*)) – Yield point equivalent to the input offset.

See also:

Engineering yield point

Notes

The point is the intersection of the curves (ϵ, σ) and $(\epsilon, E \cdot (\epsilon - \Delta\epsilon))$ being $\Delta\epsilon$ the input offset.

plot (*title*, *filePath*)
Save a figure of the stress-strain curve.

Data included in the figure:

- Stress-Strain curve.
- Elastic curve.
- Plastic curve.
- Necking curve.
- Proportionality limit point.
- Yield point.
- Ultimate point.
- Linearized elastic curve.

Parameters

- **title** (*str*) – Title for the figure.
- **filePath** (*str*) – Path to where whe figure will be saved.

plotRealCurve (*title*, *filePath*)
Save a figure of the real stress-strain curve.

Data included in the figure:

- Real stress-Strain curve.
- Real elastic curve.
- Real plastic curve.
- Real necking curve.
- Real proportionality limit point.
- Real yield point.
- Real ultimate point.
- Real linearized elastic curve.
- Hollomon's equation fitted in the elastic curve.

Parameters

- **title** (*str*) – Title for the figure.
- **filePath** (*str*) – Path to where whe figure will be saved.

saveSummaryOfProperties (*filePath*)
Save summary of the material properties to a file.

Parameters **filePath** (*str*) – Path to where the file will be saved. The file will be saved in the comma-separated-values format.

summaryOfProperties()

Summarize the material properties.

Returns **summaryOfProperties** (*pandas.DataFrame*) – Dataframe with three columns: *Property*, *Value*, *Unit*, each one with the respective material property data.

PYTHON MODULE INDEX

m

INDEX