openFDEM Post-Processing

Release 0.0

openFDEM Post-Processing

Sep 08, 2021

CONTENTS:

CHAPTER

ONE

OPENFDEM

1.1 openfdem package

1.1.1 Submodules

1.1.2 openfdem.aa_test_fdem module

1.1.3 openfdem.aggregate_storage module

class openfdem.aggregate_storage.**aggregate_storage**(*file_directory*, *h5filename=None*, *overwrite=False*, *compression=None*, *verbose=True*)

Bases: object

Aggregator class to store VTK files in a single h5 file for faster access to data. :param: none

file_group_key(*vtkfilename*)

Produces a standard group/key based on VTK file name

Parameters **vtkfilename** (*str path*) – VTK file name to be stored/read

Returns Key described using timestep and filename

Return type str

read_file(*filename*, *verbose=False*)

Extract VTK file from HDF5 file given original filename

The VTK file is reconstructed from the data arrays stored in the HDF5 file. It will be similar but different from the original.

Parameters

- **filename** (*str path*) – File name to be extracted (unaltered since HDF5 file creation)
- **verbose** (*bool, optional*) – Print progress statements, defaults to False

Returns VTK Unstructured Grid as if read from a *.vtp or *.vtu file

Return type VTK Unstructured Grid

store_file(*vtkfilename*)

Stores VTK file into HDF5 file

Parameters **vtkfilename** (*str path*) – VTK file name

1.1.4 openfdem.complete_BD_thread_pool_generators module

openfdem.complete_BD_thread_pool_generators.**history_strain_func**(*f_name*, *model*, *cv*, *ch*)

Calculate the axial stress from platens, axial strain from platens and SG as well as lateral strain from SG

Parameters

- **f_name** (*str*) – name of vtu file being processed
- **model** (openfdem.openfdem.Model) – FDEM Model Class
- **cv** (*list*) – list of cells at the corner of the vertical strain gauge
- **ch** (*list*) – list of cells at the corner of the horizontal strain gauge

Returns Stress, Platen Strain, SG Strain, SG Lateral Strain

Return type Generator[Tuple[list, list, list, list], Any, None]

openfdem.complete_BD_thread_pool_generators.**main**(*model*, *st_status*, *gauge_width*, *gauge_length*)

Main concurrent Thread Pool to calculate the full stress-strain

Parameters

- **model** (openfdem.openfdem.Model) – FDEM Model Class
- **st_status** (*bool*) – Enable/Disable SG Calculations
- **gauge_width** (*float*) – SG width
- **gauge_length** (*float*) – SG length

Returns full stress-strain information

Return type pd.DataFrame

openfdem.complete_BD_thread_pool_generators.**set_strain_gauge**(*model*, *gauge_length=None*, *gauge_width=None*)

Calculate local strain based on the dimensions of a virtual strain gauge placed at the center of teh model with x/y dimensions. By default set to 0.25 of the length/width.

Parameters

- **model** (openfdem.openfdem.Model) – FDEM Model Class
- **gauge_length** (*float*) – length of the virtual strain gauge
- **gauge_width** (*float*) – width of the virtual strain gauge

Returns Cells that cover the horizontal and vertical gauges as well as the gauge width and length

Return type [list, list, float, float]

1.1.5 openfdem.complete_UCS_thread_pool_generators module

openfdem.complete_UCS_thread_pool_generators.**history_strain_func**(*f_name*, *model*, *cv*, *ch*)

Calculate the axial stress from platens, axial strain from platens and SG as well as lateral strain from SG

Parameters

- **f_name** (*str*) – name of vtu file being processed
- **model** (openfdem.openfdem.Model) – FDEM Model Class
- **cv** (*list*) – list of cells at the corner of the vertical strain gauge
- **ch** (*list*) – list of cells at the corner of the horizontal strain gauge

Returns Stress, Platen Strain, SG Strain, SG Lateral Strain

Return type Generator[Tuple[list, list, list, list], Any, None]

openfdem.complete_UCS_thread_pool_generators.**main**(*model*, *platen_id*, *st_status*, *gauge_width*, *gauge_length*)

Main concurrent Thread Pool to calculate the full stress-strain

Parameters

- **model** (openfdem.openfdem.Model) – FDEM Model Class
- **platen_id** (*None or int*) – Manual override of Platen ID
- **st_status** (*bool*) – Enable/Disable SG Calculations
- **gauge_width** (*float*) – SG width
- **gauge_length** (*float*) – SG length

Returns full stress-strain information

Return type pd.DataFrame

openfdem.complete_UCS_thread_pool_generators.**set_strain_gauge**(*model*, *gauge_length=None*, *gauge_width=None*)

Calculate local strain based on the dimensions of a virtual strain gauge placed at the center of teh model with x/y dimensions. By default set to 0.25 of the length/width.

Parameters

- **model** (openfdem.openfdem.Model) – FDEM Model Class
- **gauge_length** (*float*) – length of the virtual strain gauge
- **gauge_width** (*float*) – width of the virtual strain gauge

Returns Cells that cover the horizontal and vertical gauges as well as the gauge width and length

Return type [list, list, float, float]

1.1.6 openfdem.extract_cell_thread_pool_generators module

openfdem.extract_cell_thread_pool_generators.**history_cellinfo_func**(*f_name*, *model*, *cell_id*, *array_needed*)

Parameters

- **f_name** (*str*) – name of vtu file being processed
- **model** (openfdem.openfdem.Model) – FDEM Model Class
- **cell_id** (*int*) – ID of the cell from which the data needs to be extracted
- **array_needed** (*str*) – Name of the property to extract

Returns The value of the property from the cell being extracted

Return type Generator[Tuple()]

openfdem.extract_cell_thread_pool_generators.**main**(*model*, *cellid*, *arrayname*)

Main concurrent Thread Pool to get value of the property from the cell being extracted

Parameters

- **model** (openfdem.openfdem.Model) – FDEM Model Class
- **cellid** (*int*) – ID of the cell from which the data needs to be extracted
- **arrayname** (*str*) – Name of the property to extract

Returns list of the values of the property from the cell being extracted

Return type list

1.1.7 openfdem.formatting_codes module

Function to calculate the time

Inputs:

- Difference in time in seconds

Returns:

- Time in minutes and seconds

openfdem.formatting_codes.**bold_text**(*val*)

openfdem.formatting_codes.**calc_timer_values**(*end_time*)

openfdem.formatting_codes.**docstring_creator**(*df*)

openfdem.formatting_codes.**green_text**(*val*)

openfdem.formatting_codes.**red_text**(*val*)

1.1.8 openfdem.model_reader module

openfdem.model_reader.**mp_read**(**args*, ***kwargs*)

openfdem.model_reader.**multiprocess_async**(**args*, ***kwargs*)

openfdem.model_reader.**multiprocess_lib_read**(**args*, ***kwargs*)

openfdem.model_reader.**normal_read**(**args*, ***kwargs*)

openfdem.model_reader.**pv_read**(**args*, ***kwargs*)

openfdem.model_reader.**pv_read_queue**(*list_of_files*, *q*)

openfdem.model_reader.**timed**(*func*)

1.1.9 openfdem.openfdem module

class openfdem.openfdem.**Model**(*folder=None*, *runfile=None*, *fdem_engine=None*)

Bases: object

Model class collects datafiles into one interface.

Each data array returns as a list ordered by timestep Collection of timesteps? handles temporal manipulations

Example

```
>>> import openfdem as fdem
>>> model = fdem.Model("../example_outputs/Irazu_UCS")
```

Eavg_mod(*ucs_data*, *upperrange*, *lowerrange*, *loc_strain='Platen Strain'*)

Average Elastic modulus between two ranges

Parameters

- **ucs_data** (*pandas.DataFrame*) – DataFrame containing the stress-strain data
- **upperrange** (*float*) – Upper range to calculate the average
- **lowerrange** (*float*) – Lower range to calculate the average
- **loc_strain** (*str*) – Column to obtain strain from. Defaults to Platen Strain

Returns Average Elastic modulus

Return type float

Raises **ZeroDivisionError** – The range over which to calculate the Eavg is too small. Consider a larger range.

Example

```
>>> data = pv.read("../example_outputs/Irazu_UCS")
>>> df_1 = data.complete_stress_strain(True)
>>> data.Eavg_mod(df_1, 0.5, 0.6)
51485.33001517835
>>> data.Eavg_mod(df_1, 0.5, 0.6, 'Gauge Displacement Y')
50976.62587224803
```

Esec_mod(*ucs_data*, *upperrange*, *loc_strain='Platen Strain'*)

Secant Modulus between 0 and upperrange. The upperrange can be a % or a fraction.

Parameters

- **ucs_data** (*pandas.DataFrame*) – DataFrame containing the stress-strain data
- **upperrange** (*float*) – Range over which to calculate the Secant Modulus
- **loc_strain** (*str*) – Column to obtain strain from. Defaults to Platen Strain

Returns Secant Elastic modulus between 0 and upperrange

Return type float

Example

```
>>> data = pv.read("../example_outputs/Irazu_UCS")
>>> df_1 = data.complete_stress_strain(True)
>>> data.Esec_mod(df_1, 0.5)
51751.010161057035
>>> data.Esec_mod(df_1, 0.5, 'Gauge Displacement Y')
51279.95421163901
```

Etan50_mod(*ucs_data*, *loc_strain='Platen Strain'*)

Tangent Elastic modulus at 50%. Calculates +/- 1 datapoint from the 50% Stress

Parameters

- **ucs_data** (*pandas.DataFrame*) – DataFrame containing the stress-strain data
- **loc_strain** (*str*) – Column to obtain strain from. Defaults to Platen Strain

Returns Tangent Elastic modulus at 50%

Return type float

Example

```
>>> data = pv.read("../example_outputs/Irazu_UCS")
>>> df_1 = data.complete_stress_strain(True)
>>> data.Etan_mod(df_1)
51539.9101160927
>>> data.Etan_mod(df_1, 'Gauge Displacement Y')
51043.327845235595
```

complete_BD_stress_strain(*st_status=False*, *gauge_width=0*, *gauge_length=0*)

Calculate the full stress-strain curve

Parameters

- **st_status** (*bool*) – Enable/Disable SG
- **gauge_length** (*float*) – length of the virtual strain gauge
- **gauge_width** (*float*) – width of the virtual strain gauge

Returns full stress-strain information

Return type pd.DataFrame

Example

```
>>> import openfdem as fdem
>>> data = fdem.Model("../example_outputs/OpenFDEM_BD")

# full stress-strain without SG
>>> df_wo_SG = data.complete_BD_stress_strain(False)
Columns:
```

(continues on next page)

(continued from previous page)

```
    Name: Platen Stress, dtype=float64, nullable: False
    Name: Platen Strain, dtype=float64, nullable: False
# full stress-strain with SG and default dimensions
>>> df_Def_SG = data.complete_BD_stress_strain(True)
Columns:
    Name: Platen Stress, dtype=float64, nullable: False
    Name: Platen Strain, dtype=float64, nullable: False
    Name: Gauge Displacement X, dtype=float64, nullable: False
    Name: Gauge Displacement Y, dtype=float64, nullable: False
# full stress-strain with SG and user-defined dimensions
>>> df_userdf_SG = data.complete_BD_stress_strain(True, 10, 10)
Columns:
    Name: Platen Stress, dtype=float64, nullable: False
    Name: Platen Strain, dtype=float64, nullable: False
    Name: Gauge Displacement X, dtype=float64, nullable: False
    Name: Gauge Displacement Y, dtype=float64, nullable: False
```

complete_stress_strain (*platen_id=None*, *st_status=False*, *gauge_width=0*, *gauge_length=0*)

Calculate the full stress-strain curve

Parameters

- **st_status** (*bool*) – Enable/Disable SG
- **platen_id** (*None or int*) – Manual override of Platen ID
- **gauge_length** (*float*) – length of the virtual strain gauge
- **gauge_width** (*float*) – width of the virtual strain gauge

Returns full stress-strain information

Return type pd.DataFrame

Example

```
>>> import openfdem as fdem
>>> data = fdem.Model("../example_outputs/Irazu_UCS")

# Minimal Arguments
>>> df_wo_SG = data.complete_stress_strain()
Columns:
    Name: Platen Stress, dtype=float64, nullable: False
    Name: Platen Strain, dtype=float64, nullable: False
# full stress-strain with SG and default dimensions
# full stress-strain without SG
>>> df_wo_SG = data.complete_stress_strain(None, False)
Columns:
    Name: Platen Stress, dtype=float64, nullable: False
    Name: Platen Strain, dtype=float64, nullable: False
# full stress-strain with SG and default dimensions
>>> df_Def_SG = data.complete_stress_strain(None, True)
Columns:
    Name: Platen Stress, dtype=float64, nullable: False
    Name: Platen Strain, dtype=float64, nullable: False
    Name: Gauge Displacement X, dtype=float64, nullable: False
    Name: Gauge Displacement Y, dtype=float64, nullable: False
# full stress-strain with SG and user-defined dimensions
>>> df_userdf_SG = data.complete_stress_strain(None, True, 10, 10)
```

(continues on next page)

(continued from previous page)

```
Columns:
    Name: Platen Stress, dtype=float64, nullable: False
    Name: Platen Strain, dtype=float64, nullable: False
    Name: Gauge Displacement X, dtype=float64, nullable: False
    Name: Gauge Displacement Y, dtype=float64, nullable: False
```

extract_cell_info(*cell_id*, *array_needed*)

find_cell(*model_point*)

Identify the nearest cell in the model to the defined point

Parameters **model_point** (*list[float, float, float]*) – x,y,z of a point in the model which

Returns the cell nearest to the point

Return type int

Example

```
>>> import openfdem as fdem
>>> data = fdem.Model("../example_outputs/Irazu_UCS")

>>> data.find_cell([0, 0, 0])
2167
>>> data.find_cell([100, 100, 0])
IndexError: Point outside model domain.
X=56.0, Y=116.0, Z=0.0
```

mat_bound_check(*mat_id*)

Checks the material ID is a valid choice.

Parameters **mat_id** (*int*) – Material ID

Returns ID of the material

Return type int

Raises **IndexError** – Material ID for platen out of range.

Example

```
>>> import openfdem as fdem
>>> model = fdem.Model("../example_outputs/Irazu_UCS")

>>> model.mat_bound_check(0)
0
>>> model.mat_bound_check(5)
IndexError: Material ID for platen out of range.
Material Range 0-1
```

model_dimensions(*mat_id=None*)

Function to get the "INITIAL" model bounds and returns the width, height, thickness

Parameters **mat_id** (*int*) – Optional, if a threshold is specific to a material type

Returns model width, model height, model thickness

Type tuple[float, float, float]

Example

```
>>> import openfdem as fdem
>>> model = fdem.Model("../example_outputs/Irazu_UCS")

# Returns the overall model dimensions
>>> model.model_dimensions
(56.0, 116.0, 0.0)
# Returns the model dimensions based on material id 1
>>> model.model_dimensions(1)
(52.0, 108.0, 0.0)
# Error when material is not found
>>> model.model_dimensions(3)
IndexError: Material ID for platen out of range.
Material Range 0-1
```

model_domain()

Identifies the model domain by confirming the simulation cell vertex. 2D (3 Points - Triangle) 3D (4 Points - Tetrahedral)

Returns number of nodes to skip in analysis

Return type int

Raises **Exception** – The simulation is not currently supported.

Example

```
>>> import openfdem as fdem
>>> model = fdem.Model("../example_outputs/Irazu_UCS")

>>> model.model_domain()
2D Simulation
4
```

openfdem_att_check(*att*)

Checks the material ID is a valid choice.

Parameters **mat_id** (*str*) – Attribute

Returns Attribute

Return type str

Raises **KeyError** – Attribute does not exist.

Example

```
>>> import openfdem as fdem
>>> model = fdem.Model("../example_outputs/Irazu_UCS")

>>> model.openfdem_att_check('material_type')
'material_type'
>>> model.openfdem_att_check('material_property')
KeyError: Attribute does not exist.
Available options are mineral_type, boundary, platen_force, platen_
↪displacement, gauge_displacement'
```

platen_info(*pv_cells*, *platen_boundary_id*, *var_property*)

This function thresholds cells based on boundary condition and sums them based on the defined parameter var_property

Parameters

- **pv_cells** (*pyvista.core.pointset.UnstructuredGrid*) –
- **platen_boundary_id** (*float*) – boundary id that the threshold should be based on
- **var_property** (*Dict[str, str]*) – name of the property (array to b returned)

Returns array of the property based on the threshold

Return type ndarray

plot_stress_strain(*strain*, *stress*, *ax=None*, ***plt_kwargs*)

Parameters

- **strain** (*pandas.DataFrame*) – X-axis data [Strain]
- **stress** (*pandas.DataFrame*) – Y-axis data [Stress]
- **ax** (*matplotlib*) – Matplotlib Axis
- **plt_kwargs** – *~matplotlib.Modules* submodules

Returns Matplotlib AxesSubplots

Return type Matplotlib Axis

Example

```
>>> import openfdem as fdem
>>> data = fdem.Model("../example_outputs/OpenFDEM_BD")

# Minimal Arguments
>>> df_wo_SG = data.complete_stress_strain()
Columns:
    Name: Platen Stress, dtype=float64, nullable: False
    Name: Platen Strain, dtype=float64, nullable: False
>>> data.plot_stress_strain(df_wo_SG['Platen Strain'], df_wo_SG[
↪'Platen Stress'], label='stress-strain', color='green')
<AxesSubplot:xlabel='Strain (-)', ylabel='Axial Stress (MPa)'>
```

rock_sample_dimensions(*platen_id=None*)

Lookup cell element ID on the top center and then trace points Using this information, we obtain the platen prop ID. Alternatively the user can define the material ID to exclude

Parameters **platen_id** (*None or int*) – Manual override of Platen ID

Returns sample width, sample height, sample thickness

Type tuple[float, float, float]

Example

```
>>> import openfdem as fdem
>>> data = fdem.Model("../example_outputs/Irazu_UCS")

# Let the script try to identify the planten material ID
>>> data.rock_sample_dimensions()
Script Identifing Platen
    Platen Material ID found as [1]
    (52.0, 108.0, 0.0)
# Explicity defined the platen material ID
>>> data.rock_sample_dimensions(0)
```

(continues on next page)

(continued from previous page)

```
User Defined Platen ID
    Platen Material ID found as [0]
    (56.0, 116.0, 0.0)
>>> data.rock_sample_dimensions(3)
IndexError: Material ID for platen out of range.
Material Range 0-1
```

simulation_type()

Identifies the type of simulation running. BD or UCS.

Returns Type of simulation. BD/UCS

Return type str

Example

```
>>> import openfdem as fdem
>>> data = fdem.Model("../example_outputs/Irazu_UCS")

# Let the script try to identify the planten material ID
>>> data.rock_sample_dimensions()
Script Identifying Platen
    Platen Material ID found as [1]
>>> data.simulation_type()
'UCS Simulation'
```

class openfdem.openfdem.**Timestep**(*file*, *runfile=None*)

Bases: object

A class handling the data of each timestep.

Each data array returns for only the timestep handles spatial manipulations

1.1.10 Module contents

CHAPTER

TWO

ABOUT OPENFDEM POST-PROCESSING MODULES

Python package for a user-friendly interaction with VTK files produced by FDEM-based codes

Project structure

- base_code: Original Python code for Irazu output post-processing. Requires Paraview installation
- docs: Documentation directory
- example_outputs: Test outputs for both base_code and the new openfdem package
- openfdem: Source folder

Github: https://github.com/OpenFDEM/OpenFDEM-Post-Processing

CHAPTER

THREE

HOW TO GET STARTED

- Navigate to Github
- Clone repository

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

O

INDEX

N

O

P

R

S

T