

Scientific Computing

Generated by Doxygen 1.8.11

Contents

1	Namespace Index	1
1.1	Packages	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Namespace Documentation	7
4.1	class_parameter Namespace Reference	7
4.1.1	Detailed Description	7
4.2	classParameter Namespace Reference	7
4.3	FEM Namespace Reference	7
4.3.1	Detailed Description	8
4.3.2	Function Documentation	8
4.3.2.1	Distribute_Force(mesh, Force)	8
4.3.2.2	get_FEM_stresses(mesh, U, param)	8
4.3.2.3	get_shapes(xi)	8
4.3.2.4	getF(mesh, Force)	8
4.3.2.5	getK(mesh, param)	9
4.3.2.6	solveSys(mesh, F, K)	9
4.4	IOlib Namespace Reference	9
4.4.1	Detailed Description	9
4.4.2	Function Documentation	10

4.4.2.1	<code>read_from_txt(fname)</code>	10
4.5	main Namespace Reference	10
4.5.1	Detailed Description	10
4.5.2	Variable Documentation	10
4.5.2.1	<code>dis_force</code>	10
4.5.2.2	<code>E</code>	10
4.5.2.3	<code>F</code>	10
4.5.2.4	<code>Force</code>	10
4.5.2.5	<code>K</code>	11
4.5.2.6	<code>mesh</code>	11
4.5.2.7	<code>nu</code>	11
4.5.2.8	<code>parameter</code>	11
4.5.2.9	<code>sig</code>	11
4.5.2.10	<code>U</code>	11
4.6	MeshDat Namespace Reference	11
4.6.1	Detailed Description	11
4.6.2	Function Documentation	11
4.6.2.1	<code>addElement(elist, ID, Enodes)</code>	11
4.6.2.2	<code>addNode(nlist, ID, coord)</code>	12
4.7	output Namespace Reference	12
4.7.1	Detailed Description	12
4.7.2	Function Documentation	12
4.7.2.1	<code>create_folder(path)</code>	12
4.7.2.2	<code>plot_solution(mesh, outfile, outfolder, U, sig)</code>	13
4.7.2.3	<code>vonMises(st)</code>	13

5	Class Documentation	15
5.1	MeshDat.Element Class Reference	15
5.1.1	Detailed Description	16
5.1.2	Constructor & Destructor Documentation	16
5.1.2.1	__init__(self, ID, parent, nodes)	16
5.1.3	Member Function Documentation	16
5.1.3.1	__getitem__(self, index)	16
5.1.3.2	__iter__(self)	16
5.1.3.3	__len__(self)	16
5.1.3.4	__str__(self)	16
5.1.3.5	get_coordinate(self, xi)	16
5.1.3.6	get_coordinates(self)	17
5.1.3.7	get_ID(self)	17
5.1.3.8	get_integration_scheme(self, name, npts)	17
5.1.3.9	get_node_IDs(self)	17
5.1.3.10	get_nodes(self)	17
5.1.3.11	get_nr_of_nodes(self)	17
5.1.3.12	get_shapes_gradient(self, xi)	17
5.2	MeshDat.Mesh Class Reference	18
5.2.1	Detailed Description	19
5.2.2	Constructor & Destructor Documentation	19
5.2.2.1	__init__(self, nodes, elems, LSnodes, RSnodes)	19
5.2.3	Member Function Documentation	19
5.2.3.1	__iter__(self)	19
5.2.3.2	__len__(self)	19
5.2.3.3	__str__(self)	19
5.2.3.4	get_elems(self)	19
5.2.3.5	get_LSnodes(self)	19
5.2.3.6	get_node(self, ID)	19
5.2.3.7	get_nodes(self)	20

5.2.3.8	get_nr_of_constraints(self)	20
5.2.3.9	get_nr_of_elements(self)	20
5.2.3.10	get_nr_of_LSnodes(self)	20
5.2.3.11	get_nr_of_nodes(self)	20
5.2.3.12	get_nr_of_nodes_with_constraints(self)	20
5.2.3.13	get_nr_of_RSnodes(self)	20
5.2.3.14	get_RSnodes(self)	20
5.3	MeshDat.Node Class Reference	21
5.3.1	Detailed Description	21
5.3.2	Constructor & Destructor Documentation	21
5.3.2.1	__init__(self, ID, coord, index)	21
5.3.3	Member Function Documentation	21
5.3.3.1	__str__(self)	22
5.3.3.2	get_constraint(self)	22
5.3.3.3	get_coordinate(self)	22
5.3.3.4	get_forcecons(self)	22
5.3.3.5	get_ID(self)	22
5.3.3.6	get_index(self)	22
5.3.3.7	set_constraint(self, cons)	22
5.3.3.8	set_coordinate(self, coord)	22
5.3.3.9	set_forcecons(self, consF)	23
5.4	classParameter.Parameter Class Reference	23
5.4.1	Detailed Description	23
5.4.2	Constructor & Destructor Documentation	23
5.4.2.1	__init__(self, E, nu)	23
5.4.3	Member Data Documentation	23
5.4.3.1	E	24
5.4.3.2	nu	24
5.5	MeshDat.StandardTriangle Class Reference	24
5.5.1	Detailed Description	24
5.5.2	Member Function Documentation	24
5.5.2.1	__len__(self)	24
5.5.2.2	get_connections_scheme(self)	24
5.5.2.3	get_integration_scheme(self, name, npts)	24
5.5.2.4	get_nr_of_nodes(self)	25
5.5.2.5	get_shapes(self, xi)	25
5.5.2.6	get_shapes_gradient(self, xi)	25

6	File Documentation	27
6.1	classParameter.py File Reference	27
6.2	FEM.py File Reference	27
6.3	IOlib.py File Reference	28
6.4	main.py File Reference	28
6.5	MeshDat.py File Reference	28
6.6	output.py File Reference	29

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

class_parameter	This file contains all material parameters	7
classParameter	7
FEM	This module contains the functions used for the FEM	7
IOlib	This file is used to read the mesh from a .msh file	9
main	This file is used to run the script and call al functions	10
MeshDat	This file contains all objects	11
output	This module contains the functions used for creating the output	12

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

MeshDat.Element	
Element object	15
MeshDat.Mesh	
Finite element mesh	18
MeshDat.Node	
Node object	21
classParameter.Parameter	
Parameter object	23
MeshDat.StandardTriangle	
Linear triangle parent element with local coordinates (0,0), (1,0), (0,1)	24

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

classParameter.py	27
FEM.py	27
IOlib.py	28
main.py	28
MeshDat.py	28
output.py	29

Chapter 4

Namespace Documentation

4.1 `class_parameter` Namespace Reference

This file contains all material parameters.

4.1.1 Detailed Description

This file contains all material parameters.

4.2 `classParameter` Namespace Reference

Classes

- class `Parameter`
`Parameter` object.

4.3 FEM Namespace Reference

This module contains the functions used for the `FEM`.

Functions

- def `Distribute_Force` (mesh, Force)
Distribute the Force over the nodes on the righthand side.
- def `getF` (mesh, Force)
Limit F to f where the entries of K in $KU=F$ are known.
- def `solveSys` (mesh, F, K)
Rewrite K to K_n (rows with known U are removed) and solve the linear system.
- def `getK` (mesh, param)
Make stiffness matrix K from all separate K_e from the elements.
- def `get_FEM_stresses` (mesh, U, param)
returns a array of stresses in all elements in $[xx\ yy\ xy]$
- def `get_shapes` (xi)

4.3.1 Detailed Description

This module contains the functions used for the [FEM](#).

4.3.2 Function Documentation

4.3.2.1 `def FEM.Distribute_Force (mesh, Force)`

Distribute the Force over the nodes on the righthand side.

Parameters

<i>mesh</i>	This is a MeshDat.Mesh()
<i>Force</i>	Force on the nodes on the righthand side

Returns

Distributed force on nodes

4.3.2.2 `def FEM.get_FEM_stresses (mesh, U, param)`

returns a array of stresses in all elements in [xx yy xy]

Parameters

<i>mesh</i>	This is a MeshDat.Mesh()
<i>U</i>	Displacement of nodes
<i>param</i>	This is a <code>class_parameter.Parameter()</code>

Returns

Stiffness K

4.3.2.3 `def FEM.get_shapes (xi)`

4.3.2.4 `def FEM.getF (mesh, Force)`

Limit F to f where the entries of K in $KU=F$ are known.

Parameters

<i>mesh</i>	This is a MeshDat.Mesh()
<i>Force</i>	Distributed force on nodes

Returns

Limited F

4.3.2.5 def FEM.getK (mesh, param)

Make stiffness matrix K from all separate Ke from the elements.

Parameters

<i>mesh</i>	This is a MeshDat.Mesh()
<i>param</i>	This is a <code>class_parameter.Parameter()</code>

Returns

Stifness K

4.3.2.6 def FEM.solveSys (mesh, F, K)

Rewrite K to Kn (rows with known U are removed) and solve the linear system.

Parameters

<i>mesh</i>	This is a MeshDat.Mesh()
<i>F</i>	Limited Force Distribution
<i>K</i>	Stiffness Matrix

Returns

Displacement U

4.4 IOlib Namespace Reference

This file is used to read the mesh from a .msh file.

Functions

- def [read_from_txt](#) (fname)
Mesh file reader.

4.4.1 Detailed Description

This file is used to read the mesh from a .msh file.

`-*- coding: utf-8 -*-`

4.4.2 Function Documentation

4.4.2.1 `def IOlib.read_from_txt (fname)`

Mesh file reader.

Parameters

<i>fname</i>	Name of the mesh file
--------------	-----------------------

Returns

Finite element mesh

4.5 main Namespace Reference

This file is used to run the script and call al functions.

Variables

- int `E` = 70
- float `nu` = 0.33
- `parameter` = `Parameter(E, nu)`
- `mesh` = `read_from_txt('Mesh/MultiHole.msh')`
- int `Force` = -1
- `dis_force` = `Distribute_Force(mesh,Force)`
- `K` = `getK(mesh,parameter)`
- `F` = `getF(mesh,dis_force)`
- `U` = `solveSys(mesh,F,K)`
- `sig` = `get_FEM_stresses(mesh,U,parameter)`

4.5.1 Detailed Description

This file is used to run the script and call al functions.

4.5.2 Variable Documentation

4.5.2.1 `main.dis_force = Distribute_Force(mesh,Force)`

4.5.2.2 `int main.E = 70`

4.5.2.3 `main.F = getF(mesh,dis_force)`

4.5.2.4 `int main.Force = -1`

4.5.2.5 `main.K = getK(mesh,parameter)`

4.5.2.6 `main.mesh = read_from_txt('Mesh/MultiHole.msh')`

4.5.2.7 `float main.nu = 0.33`

4.5.2.8 `main.parameter = Parameter(E, nu)`

4.5.2.9 `main.sig = get_FEM_stresses(mesh,U,parameter)`

4.5.2.10 `main.U = solveSys(mesh,F,K)`

4.6 MeshDat Namespace Reference

This file contains all objects.

Classes

- class [Element](#)
element object
- class [Mesh](#)
Finite element mesh.
- class [Node](#)
node object
- class [StandardTriangle](#)
Linear triangle parent element with local coordinates (0,0), (1,0), (0,1)

Functions

- def [addNode](#) (nlist, ID, coord)
Add nodes to a list before creating the mesh.
- def [addElement](#) (elist, ID, Enodes)
Add elements to a list before creating the mesh.

4.6.1 Detailed Description

This file contains all objects.

4.6.2 Function Documentation

4.6.2.1 `def MeshDat.addElement (elist, ID, Enodes)`

Add elements to a list before creating the mesh.

Parameters

<i>elist</i>	List of elements
<i>ID</i>	Element ID
<i>Enodes</i>	Element nodes

4.6.2.2 def MeshDat.addNode (*nlist*, *ID*, *coord*)

Add nodes to a list before creating the mesh.

Parameters

<i>nlist</i>	List of nodes
<i>ID</i>	Node ID
<i>coord</i>	Node coordinate

4.7 output Namespace Reference

This module contains the functions used for creating the output.

Functions

- def [plot_solution](#) (mesh, outfile, outfolder, U, sig)
plot_solution.
- def [create_folder](#) (path)
create_folder Creates output Folder if it does not exist already
- def [vonMises](#) (st)
vonMises Calculates the Von Mises stress on elements

4.7.1 Detailed Description

This module contains the functions used for creating the output.

The ouput is formatted in .VTK and can be opened with Paraview.

4.7.2 Function Documentation**4.7.2.1 def output.create_folder (*path*)**

create_folder Creates output Folder if it does not exist already

Parameters

<i>path</i>	defines the output path.
-------------	--------------------------

4.7.2.2 def output.plot_solution (*mesh*, *outfile*, *outfolder*, *U*, *sig*)

plot_solution.

Plots the nodes, elements, stresses and displacements of the mesh

Parameters

<i>mesh</i>	This is a MeshDat.Mesh()
<i>outfile</i>	Name of the outputfile
<i>outfolder</i>	Location of the outputfile
<i>U</i>	In U the displacements of the nodes are stated
<i>sig</i>	In sig the stresses for all nodes are listed

4.7.2.3 def output.vonMises (*st*)

vonMises Calculates the Von Mises stress on elements

Parameters

<i>st</i>	Vector containing the stress elements
-----------	---------------------------------------

Chapter 5

Class Documentation

5.1 MeshDat.Element Class Reference

element object

Public Member Functions

- def `__init__` (self, ID, parent, nodes)
Constructor.
- def `get_ID` (self)
Get element ID.
- def `get_nodes` (self)
Get nodes.
- def `get_node_IDs` (self)
Get node IDs from element.
- def `get_nr_of_nodes` (self)
Get the number of nodes.
- def `__iter__` (self)
Iterator function.
- def `__len__` (self)
Length function.
- def `__str__` (self)
String function.
- def `__getitem__` (self, index)
Get item function.
- def `get_integration_scheme` (self, name, npts)
Get the integration scheme.
- def `get_shapes_gradient` (self, xi)
Get the shape functions gradient.
- def `get_coordinates` (self)
Get the matrix of nodal coordinates.
- def `get_coordinate` (self, xi)
Get the global coordinate.

5.1.1 Detailed Description

element object

5.1.2 Constructor & Destructor Documentation

5.1.2.1 `def MeshDat.Element.__init__(self, ID, parent, nodes)`

Constructor.

Parameters

<i>ID</i>	Element ID
<i>parent</i>	Standard/parent element
<i>nodes</i>	List of finite element Nodes

5.1.3 Member Function Documentation

5.1.3.1 `def MeshDat.Element.__getitem__(self, index)`

Get item function.

5.1.3.2 `def MeshDat.Element.__iter__(self)`

Iterator function.

5.1.3.3 `def MeshDat.Element.__len__(self)`

Length function.

5.1.3.4 `def MeshDat.Element.__str__(self)`

String function.

5.1.3.5 `def MeshDat.Element.get_coordinate(self, xi)`

Get the global coordinate.

Parameters

<i>xi</i>	Local coordinate vector
-----------	-------------------------

Returns

Global coordinate vector

5.1.3.6 `def MeshDat.Element.get_coordinates (self)`

Get the matrix of nodal coordinates.

5.1.3.7 `def MeshDat.Element.get_ID (self)`

Get element ID.

5.1.3.8 `def MeshDat.Element.get_integration_scheme (self, name, npts)`

Get the integration scheme.

Parameters

<i>name</i>	The type of integration scheme (e.g. 'gauss')
<i>npts</i>	The number of integration points

Returns

Matrix of integration point coordinates
Vector of integration point weights

5.1.3.9 `def MeshDat.Element.get_node_IDs (self)`

Get node IDs from element.

5.1.3.10 `def MeshDat.Element.get_nodes (self)`

Get nodes.

5.1.3.11 `def MeshDat.Element.get_nr_of_nodes (self)`

Get the number of nodes.

5.1.3.12 `def MeshDat.Element.get_shapes_gradient (self, xi)`

Get the shape functions gradient.

Parameters

xi	Local coordinate vector
------	-------------------------

Returns

Matrix of shape function gradients

The documentation for this class was generated from the following file:

- [MeshDat.py](#)

5.2 MeshDat.Mesh Class Reference

Finite element mesh.

Public Member Functions

- def `__init__` (self, nodes, elems, LSnodes, RSnodes)
Constructor.
- def `get_node` (self, ID)
Get a node.
- def `__iter__` (self)
Iterator function.
- def `__len__` (self)
Length function.
- def `__str__` (self)
String function.
- def `get_nodes` (self)
Get the list of nodes in the mesh.
- def `get_elems` (self)
Get the list of elements in the mesh.
- def `get_LSnodes` (self)
Get the list of nodes on the left boundary of the mesh.
- def `get_RSnodes` (self)
Get the list of nodes on the right boundary of the mesh.
- def `get_nr_of_nodes` (self)
Get the number of nodes.
- def `get_nr_of_elements` (self)
Get the number of elements.
- def `get_nr_of_LSnodes` (self)
Get the number of boundary nodes.
- def `get_nr_of_RSnodes` (self)
Get the number of boundary nodes.
- def `get_nr_of_constraints` (self)
Get the number of constrains.
- def `get_nr_of_nodes_with_constraints` (self)
Get the number of constrained nodes.

5.2.1 Detailed Description

Finite element mesh.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 `def MeshDat.Mesh.__init__(self, nodes, elems, LSnodes, RSnodes)`

Constructor.

Parameters

<i>nodes</i>	list of finite element nodes
<i>elems</i>	list of finite elements
<i>LSnodes</i>	list of boundary nodes on left side of geometry
<i>RSnodes</i>	list of boundary nodes on right side of geometry

5.2.3 Member Function Documentation

5.2.3.1 `def MeshDat.Mesh.__iter__(self)`

Iterator function.

5.2.3.2 `def MeshDat.Mesh.__len__(self)`

Length function.

5.2.3.3 `def MeshDat.Mesh.__str__(self)`

String function.

5.2.3.4 `def MeshDat.Mesh.get_elems(self)`

Get the list of elements in the mesh.

5.2.3.5 `def MeshDat.Mesh.get_LSnodes(self)`

Get the list of nodes on the left boundary of the mesh.

5.2.3.6 `def MeshDat.Mesh.get_node(self, ID)`

Get a node.

Parameters

<i>ID</i>	Node ID
-----------	-------------------------

5.2.3.7 `def MeshDat.Mesh.get_nodes (self)`

Get the list of nodes in the mesh.

5.2.3.8 `def MeshDat.Mesh.get_nr_of_constraints (self)`

Get the number of constrains.

5.2.3.9 `def MeshDat.Mesh.get_nr_of_elements (self)`

Get the number of elements.

5.2.3.10 `def MeshDat.Mesh.get_nr_of_LSnodes (self)`

Get the number of boundary nodes.

5.2.3.11 `def MeshDat.Mesh.get_nr_of_nodes (self)`

Get the number of nodes.

5.2.3.12 `def MeshDat.Mesh.get_nr_of_nodes_with_constraints (self)`

Get the number of constrained nodes.

5.2.3.13 `def MeshDat.Mesh.get_nr_of_RSnodes (self)`

Get the number of boundary nodes.

5.2.3.14 `def MeshDat.Mesh.get_RSnodes (self)`

Get the list of nodes on the right boundary of the mesh.

The documentation for this class was generated from the following file:

- [MeshDat.py](#)

5.3 MeshDat.Node Class Reference

node object

Public Member Functions

- def `__init__` (self, ID, coord, index)
Constructor.
- def `get_ID` (self)
Get the [Node](#) ID.
- def `get_index` (self)
Get the [Node](#) index.
- def `set_coordinate` (self, coord)
Set the coordinate.
- def `get_coordinate` (self)
Get the [Node](#) coordinate.
- def `__str__` (self)
String function.
- def `set_constraint` (self, cons)
Set Constraint.
- def `get_constraint` (self)
Get Constraint.
- def `set_forcecons` (self, consF)
Set Force Constraint.
- def `get_forcecons` (self)
Get ForceConstraint.

5.3.1 Detailed Description

node object

5.3.2 Constructor & Destructor Documentation

5.3.2.1 def MeshDat.Node.__init__ (self, ID, coord, index)

Constructor.

Parameters

<i>ID</i>	Node ID
<i>coord</i>	Node coordinate
<i>index</i>	Node index

5.3.3 Member Function Documentation

5.3.3.1 `def MeshDat.Node.__str__(self)`

String function.

5.3.3.2 `def MeshDat.Node.get_constraint(self)`

Get Constraint.

5.3.3.3 `def MeshDat.Node.get_coordinate(self)`

Get the [Node](#) coordinate.

5.3.3.4 `def MeshDat.Node.get_forcecons(self)`

Get ForceConstraint.

5.3.3.5 `def MeshDat.Node.get_ID(self)`

Get the [Node](#) ID.

5.3.3.6 `def MeshDat.Node.get_index(self)`

Get the [Node](#) index.

5.3.3.7 `def MeshDat.Node.set_constraint(self, cons)`

Set Constraint.

Parameters

<i>cons</i>	Set constraint for node
-------------	-------------------------

5.3.3.8 `def MeshDat.Node.set_coordinate(self, coord)`

Set the coordinate.

Parameters

<i>coord</i>	Node coordinate
--------------	---------------------------------

5.3.3.9 `def MeshDat.Node.set_forcecons (self, consF)`

Set Force Constraint.

Parameters

<i>consF</i>	set Force constraints
--------------	-----------------------

The documentation for this class was generated from the following file:

- [MeshDat.py](#)

5.4 classParameter.Parameter Class Reference

[Parameter](#) object.

Public Member Functions

- `def __init__ (self, E, nu)`
Constructor.

Public Attributes

- [E](#)
- [nu](#)

5.4.1 Detailed Description

[Parameter](#) object.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 `def classParameter.Parameter.__init__ (self, E, nu)`

Constructor.

Parameters

<i>E</i>	= Young's Modulus [Pa]
<i>nu</i>	= Poisson Ratio []

5.4.3 Member Data Documentation

5.4.3.1 classParameter.Parameter.E

5.4.3.2 classParameter.Parameter.nu

The documentation for this class was generated from the following file:

- [classParameter.py](#)

5.5 MeshDat.StandardTriangle Class Reference

Linear triangle parent element with local coordinates (0,0), (1,0), (0,1)

Public Member Functions

- def [__len__](#) (self)
Length function.
- def [get_nr_of_nodes](#) (self)
Get the number of nodes.
- def [get_shapes](#) (self, xi)
Get the shape functions.
- def [get_shapes_gradient](#) (self, xi)
Get the shape functions gradient.
- def [get_integration_scheme](#) (self, name, npts)
Get the integration scheme.
- def [get_connections_scheme](#) (self)
Get the element's internal connection scheme.

5.5.1 Detailed Description

Linear triangle parent element with local coordinates (0,0), (1,0), (0,1)

5.5.2 Member Function Documentation

5.5.2.1 def MeshDat.StandardTriangle.__len__ (self)

Length function.

5.5.2.2 def MeshDat.StandardTriangle.get_connections_scheme (self)

Get the element's internal connection scheme.

5.5.2.3 def MeshDat.StandardTriangle.get_integration_scheme (self, name, npts)

Get the integration scheme.

Parameters

<i>name</i>	The type of integration scheme (e.g. 'gauss')
<i>npts</i>	The number of integration points

Returns

Matrix of integration point coordinates
Vector of integration point weights

5.5.2.4 `def MeshDat.StandardTriangle.get_nr_of_nodes (self)`

Get the number of nodes.

5.5.2.5 `def MeshDat.StandardTriangle.get_shapes (self, xi)`

Get the shape functions.

Parameters

<i>xi</i>	Local coordinate vector
-----------	-------------------------

Returns

Vector of shape functions

5.5.2.6 `def MeshDat.StandardTriangle.get_shapes_gradient (self, xi)`

Get the shape functions gradient.

Parameters

<i>xi</i>	Local coordinate vector
-----------	-------------------------

Returns

Matrix of shape function gradients

The documentation for this class was generated from the following file:

- [MeshDat.py](#)

Chapter 6

File Documentation

6.1 classParameter.py File Reference

Classes

- class [classParameter.Parameter](#)
Parameter object.

Namespaces

- [classParameter](#)
- [class_parameter](#)
This file contains all material parameters.

6.2 FEM.py File Reference

Namespaces

- [FEM](#)
This module contains the functions used for the [FEM](#).

Functions

- def [FEM.Distribute_Force](#) (mesh, Force)
Distribute the Force over the nodes on the righthand side.
- def [FEM.getF](#) (mesh, Force)
Limit F to f where the entries of K in $KU=F$ are known.
- def [FEM.solveSys](#) (mesh, F, K)
Rewrite K to K_n (rows with known U are removed) and solve the linear system.
- def [FEM.getK](#) (mesh, param)
Make stiffness matrix K from all separate K_e from the elements.
- def [FEM.get_FEM_stresses](#) (mesh, U, param)
returns a array of stresses in all elements in [xx yy xy]
- def [FEM.get_shapes](#) (xi)

6.3 IOlib.py File Reference

Namespaces

- [IOlib](#)

This file is used to read the mesh from a .msh file.

Functions

- def [IOlib.read_from_txt](#) (fname)

Mesh file reader.

6.4 main.py File Reference

Namespaces

- [main](#)

This file is used to run the script and call all functions.

Variables

- int [main.E](#) = 70
- float [main.nu](#) = 0.33
- [main.parameter](#) = Parameter(E, nu)
- [main.mesh](#) = read_from_txt('Mesh/MultiHole.msh')
- int [main.Force](#) = -1
- [main.dis_force](#) = Distribute_Force(mesh, Force)
- [main.K](#) = getK(mesh, parameter)
- [main.F](#) = getF(mesh, dis_force)
- [main.U](#) = solveSys(mesh, F, K)
- [main.sig](#) = get_FEM_stresses(mesh, U, parameter)

6.5 MeshDat.py File Reference

Classes

- class [MeshDat.Node](#)

node object

- class [MeshDat.Element](#)

element object

- class [MeshDat.Mesh](#)

Finite element mesh.

- class [MeshDat.StandardTriangle](#)

Linear triangle parent element with local coordinates (0,0), (1,0), (0,1)

Namespaces

- [MeshDat](#)

This file contains all objects.

Functions

- def [MeshDat.addNode](#) (nlist, ID, coord)
Add nodes to a list before creating the mesh.
- def [MeshDat.addElement](#) (elist, ID, Enodes)
Add elements to a list before creating the mesh.

6.6 output.py File Reference

Namespaces

- [output](#)

This module contains the functions used for creating the output.

Functions

- def [output.plot_solution](#) (mesh, outfile, outfolder, U, sig)
plot_solution.
- def [output.create_folder](#) (path)
create_folder Creates output Folder if it does not exist already
- def [output.vonMises](#) (st)
vonMises Calculates the Von Mises stress on elements

