Scientific Computing

Generated by Doxygen 1.8.11

Contents

Chapter 1

Namespace Index

1.1 Packages

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

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 Kn (rows with known U are removed) and solve the linear system.
- def getK (mesh, param)

  Make stiffness matrix K from all separate Ke 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

mesh	This is a MeshDat.Mesh()
Force	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

mesh	This is a MeshDat.Mesh()
U	Displacement of nodes
param	This is a class_parameter.Parameter()

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

mesh	This is a MeshDat.Mesh()
Force	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

mesh	This is a MeshDat.Mesh()
param	This is a class_parameter.Parameter()

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

mesh	This is a MeshDat.Mesh()
F	Limited Force Distribution
K	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

fname	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

elist	List of elements
ID	Element ID
Enodes	Element nodes

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

Add nodes to a list before creating the mesh.

Parameters

nlist	List of nodes
ID	Node ID
coord	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

path	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

mesh	This is a MeshDat.Mesh()
outfile	Name of the outputfile
outfolder	Location of the outputfile
U	In U the displacements of the nodes are stated
sig	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

st	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

ID	Element ID
parent	Standard/parent element
nodes	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

xi	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

name	The type of integration scheme (e.g. 'gauss')
npts	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

nodes	list of finite element nodes
elems	list of finite elements
LSnodes	list of boundary nodes on left side of geometry
RSnodes	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

ID	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

ID	Node ID
coord	Node coordinate
index	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

cons	Set constraint for node

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

Set the coordinate.

Parameters

coord	Node coordinate

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

Set Force Constraint.

Parameters

consF	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

E	= Young's Modulus [Pa]
nu	= 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

name	The type of integration scheme (e.g. 'gauss')
npts	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

xi	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

xi	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 Kn (rows with known U are removed) and solve the linear system.
- def FEM.getK (mesh, param)

  Make stiffness matrix K from all separate Ke 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 al 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