

---

# **TeeFEM - Teekkari FEM Documentation**

*Release 0.1.0*

**Jukka Aho**

March 26, 2012



# CONTENTS

<b>1</b>	<b>Indices and tables</b>	<b>3</b>
1.1	### Elements.py - FINITE ELEMENTS ### . . . . .	8
1.2	### materials.py - MATERIAL MODELS ### . . . . .	9
1.3	Discrete Kirchhoff elements . . . . .	16
	<b>Module Index</b>	<b>21</b>
	<b>Index</b>	<b>23</b>



Contents:



# INDICES AND TABLES

- *Index*
- *Module Index*
- *Search Page*

Created on Wed Mar 07 23:12:23 2012

@author: Jukka Aho

Mesh routines

**class Mesh** (\*args, \*\*kws)

Bases: object

Mesh class. Reads file formats: Gmsh .msh, Salome .med, Aster .mail

**assign\_material** (\*args, \*\*kwargs)

**create\_node\_groups** ()

Create node groups from element groups

**find\_group\_ma** (gma)

**find\_group\_no** (gno)

**status** (fulloutput=False)

**class Net** (\*args, \*\*kws)

Bases: object

**class Net1D** (\*args, \*\*kws)

Bases: teefem.geom.Net

1d shapes

**class Net2D** (\*args, \*\*kws)

Bases: teefem.geom.Net

2d shapes

**J** (\*ke)

2d Jacobian matrix

**detJ** (\*ke)

Jacobian determinant

**invJ** (\*ke)

Returns inverse Jacobian matrix

**plot** (\*\*kws)

Plot geometry

**plot3d** (\*\*kws)

Plot geometry with mplot3d

```
class Node (x=0, y=0, z=0, **kws)
```

Bases: object

Node object

**INPUT:**

- x
- y
- z

**OUTPUT:**

- Node object

**EXAMPLES:**

```
>>> n = Node(x = 1, y = 2, z = 3)
>>> print n
Node : (1.00, 2.00, 3.00)
```

**assign\_boundary\_condition** (bc)

**hasbc** (dof)

**hasload** (dof)

**status** ()

**update\_boundary\_conditions** ()

**update\_field** (\*args, \*\*kws)

**valbc** (dof)

**valload** (dof)

```
class Poi1 (*args, **kws)
```

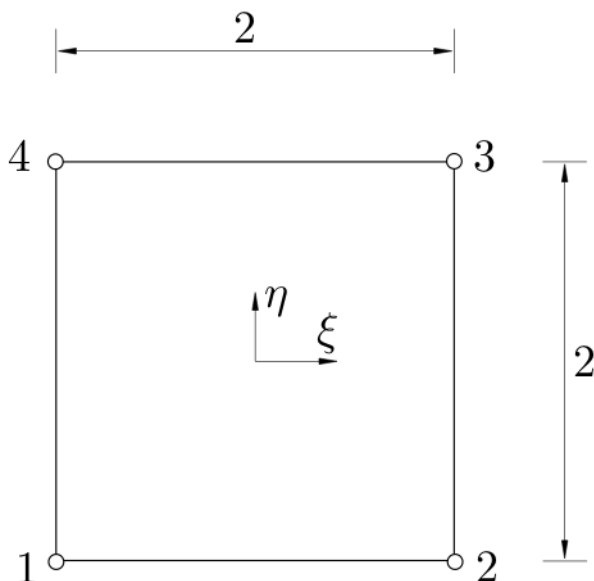
Bases: teefem.geom.Net1D

1-node point.

```
class Quad4 (*args, **kws)
```

Bases: teefem.geom.Net2D

4-node quadrangle shape





SHAPE FUNCTION:

INPUT:

OUTPUT:

EXAMPLES:

```
class Quad8 (*args, **kws)
```

Bases: `teefem.geom.Net2D`

8-node second order quadrangle (4 nodes associated with the vertices and 4 with the edges).

```
plot ()
```

```
class Quad9 (*args, **kws)
```

Bases: `teefem.geom.Net2D`

9-node second order quadrangle (4 nodes associated with the vertices, 4 with the edges and 1 with the face).

```
class Seg2 (*args, **kws)
```

Bases: `teefem.geom.Net1D`

2-node line.

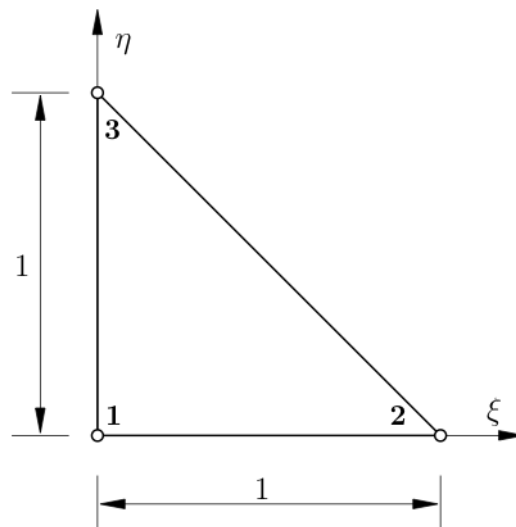
```
class Seg3 (*args, **kws)
```

Bases: `teefem.geom.Net1D`

3-node second order line (2 nodes associated with the vertices and 1 with the edge).

```
class Tri3 (*args, **kws)
```

Bases: `teefem.geom.Net2D`



**3-node triangle.**

SHAPE FUNCTIONS:

INPUT:

- name - Name of shape
- [nodes] - List of nodes (`len(nodes)==3`)

OUTPUT:

- Tri3 object

EXAMPLES:

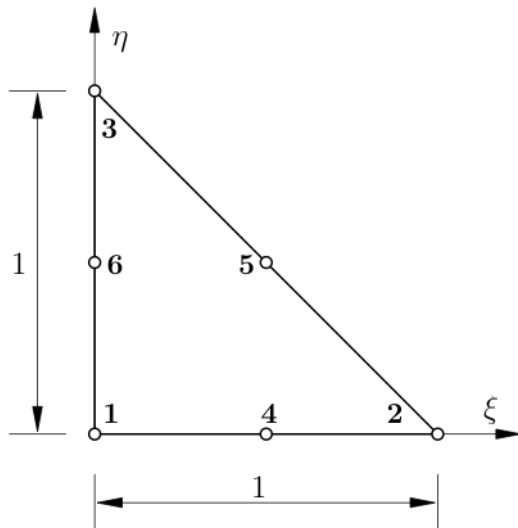
```
>>> n1 = Node(x=0.0, y=0.0)
>>> n2 = Node(x=2.0, y=1.0)
```

```
>>> n3 = Node(x=0.5, y=2.0)
>>> tr3 = Tria3(name = 'tr3', nodes = [n1, n2, n3])
```

**class Tria6** (\*args, \*\*kws)

Bases: `teefem.geom.Net2D`

6-node second order triangle (3 nodes associated with the vertices and 3 with the edges).



SHAPE FUNCTIONS:

INPUT:

- name - Name of shape
- [nodes] - List of nodes (`len(nodes)==6`)

OUTPUT:

- Tria6 object

EXAMPLES:

```
>>> n1 = Node(x=0.0, y=0.0)
>>> n2 = Node(x=2.0, y=1.0)
>>> n3 = Node(x=0.5, y=2.0)
>>> n4 = Node(x=1.0, y=0.0)
>>> n5 = Node(x=1.0, y=1.0)
>>> n6 = Node(x=0.2, y=1.0)
>>> tr6 = Tria6(name = 'tr6', nodes = [n1, n2, n3, n4, n5, n6])
```

**mesh** (\*\*kws)

**parsemail** (fh)

Parse Aster mail format

**parsemed** (fh)

Read Salome .med file format

**parsemsh** (fh)

Read Gmsh .msh file format

Created on Tue Mar 20 10:38:43 2012

@author: Jukka

**class BoundaryCondition** ()

Bases: `object`

```
class DirichletBoundaryCondition (**kws)
    Bases: teefem.boundary_conditions.BoundaryCondition
    exist (dof)
    val (dof)
```

```
class NodalForce (**kws)
    Bases: teefem.boundary_conditions.BoundaryCondition
    exist (dof)
    val (dof)
```

```
class PressureBoundaryCondition (**kws)
    Bases: teefem.boundary_conditions.BoundaryCondition
    INPUT KEYWORDS
```

```
        pressure : function f.e. pressure = lambda k,e: -100e3
```

```
dirichlet_bc
    alias of DirichletBoundaryCondition
```

```
nodal_force
    alias of NodalForce
```

```
pressure_bc
    alias of PressureBoundaryCondition
```

```
test1 ()
```

Created on Wed Mar 14 14:20:39 2012

@author: Jukka Aho

Cache

#### **Cache**

Decorator that caches a function's return value each time it is called. If called later with the same arguments, the cached value is returned, and not re-evaluated.

#### **cache**

Decorator that caches a function's return value each time it is called. If called later with the same arguments, the cached value is returned, and not re-evaluated.

#### **fibonacci**

Decorator that caches a function's return value each time it is called. If called later with the same arguments, the cached value is returned, and not re-evaluated.

Created on Sun Mar 11 02:04:59 2012

@author: Jukka

```
class IJV ()
    Bases: object
    Creates IJV-arrays and converts to COO sparse matrix
```

#### **INPUT:**

- dim - lenght of I,J,V lists

#### **EXAMPLES:**

```
>>> k = 210e9*10e-4/3
>>> ijv = IJV()
>>> ijv.add(0,0,k)
>>> ijv.add(1,1,k)
>>> ijv.add(0,1,-k)
>>> ijv.add(1,0,-k)
```

```
>>> print i.jv.tocoo().todense()
[[ 70000000. -70000000.]
 [-70000000.  70000000.]]

add(i, j, v)

tocoo()

anglxy(n1, n2)

arrtostr(arr)
    Converts list of integers to string

dist(n1, n2)
    Returns distance between two nodes

test1()

Created on Tue Mar 20 07:47:37 2012

DeprecationWarning decorator

deprecated(func)
    This is a decorator which can be used to mark functions as deprecated. It will result in a warning being
    emitted when the function is used.

Created on Thu Mar 08 00:03:31 2012

@author: Jukka Aho
```

## 1.1 **### Elements.py - FINITE ELEMENTS ###**

```
class Element(*args, **kws)
    Bases: object

    General finite element class

    J

    area

    invJ

    status

    update(U)

class Element1D(*args, **kws)
    Bases: teefem.elements.Element

    General functions for 1D finite elements

class Element2D(*args, **kws)
    Bases: teefem.elements.Element

    General 2D finite element class

    plot(*args, **kws)
        Plots geometry and Gauss points
```

Created on Sun Feb 12 16:11:41 2012

@author: Jukka Aho      Created on Sun Feb 12 16:45:50 2012

@author: Jukka Aho

## 1.2 ### materials.py - MATERIAL MODELS ###

**class Elastic** (\*args, \*\*kws)

Bases: object

Linear elastic material model

INPUT:

- E - Modulus of elasticity
- nu - Poisson's ratio

OUTPUT:

- Material object

EXAMPLES:

```
>>> mesh = Mesh(...) # doctest: +SKIP
>>> mat = Elastic(E = 210e9, nu=0.3)
>>> mesh.assign_material(mat) # doctest: +SKIP
```

**elastic**

alias of Elastic

Created on Tue Mar 20 17:32:59 2012

@author: Jukka Aho

Mesh tools

Erilaisia yksinkertaisia verkotustyökaluja joilla voi muodostaa perusverkkoja testitarkoituksiin.

Mikään näistä ei toimi. Tästä voisi yrittää raapia kasaan jotakin toimivaa. Perusajatus olisi että esim.

```
>>> import teefem
>>> mesh1 = teefem.mesh.unitcircle(r=1)
>>> print mesh1
```

**class Trimesh** (x, y)

Bases: matplotlib.delaunay.triangulate.Triangulation

Kolmioverkkorutiini matplotlibin kirjastosta paranneltuna

**average\_edge\_lenght**

**plot** ()

**sgm** (fixedn)

Ei toimi.

**force\_eq** (mesh, fd, fh, plot=True, filename='output/force\_eq')

Tasapainottaa pistejoukon voimatasapainolla käyttäen kolmiointia.

**force\_eq\_test1** ()

**forcemoc** (x0, y0, plot=True, filename='moc')

Tasapainottaa verkon voimatasapainolla. Tämä versio perustuu siihen että kaikki partikkelit vaikuttavat kaikkiin partikkeleihin. Tässä versiossa ei siis tarvitse kolmiointia tai muutakaan.

**forcemoc\_test1** ()

**refine** (mesh, \*\*kws)

Yksinkertainen refine looppi. Katkaistaan liian pitkät reunat keskeltä.

**sgmtest** ()

Testataan SGM rutiinia

**test1** ()

**unitcircle** ( $R=1$ ,  $lc=1$ )

Yksikköympyrä

**unitcircle1** ( $R=1$ ,  $lc=0.5$ )

Yksikköympyrä

**unitcircle2** ( $R=1$ ,  $lc=1.0$ )

Toinen versio yksikköympyrästä, alkujakauma kuten distmeshissä

**unitinterval** ( $a=0$ ,  $b=1$ ,  $lc=0.10000000000000001$ )

**unitsquare** ( $a=1$ ,  $b=1$ ,  $lc=0.20000000000000001$ )

Suorakaiteen muotoinen alue

**unitsquare2** ( $L=1$ ,  $H=1$ )

Suorakaiteen muotoinen yksinkertainen käsin tehty verkotus

P4——GA4——P3 |||| OM1 ||||

GA1 P5 GA3 |

|  
|  
|  
|

P1——GA2——P2

Created on Sat Mar 24 02:27:29 2012

@author: Jukka

Kaikenlaisia verkotukseen liittyviä kokeiluja

**arctan2** ( $y$ ,  $x$ )

**dist** ( $x1$ ,  $x2$ )

**onedmeshtest** ()

Yksidimensiainen verkon tasapainotustesti

Perustuu suuriin siirtymiin ja staattiseen analyysiin

**onedmeshtest2** ()

Yksidimensiainen verkon tasapainotustesti  $mu'' + cu' + ku' = F$

$x0|—x2——|x1$

**onedmeshtest3** ()

Yksidimensiainen verkon tasapainotustesti  $mu'' + cu' + ku' = F$

$x0|—x2——|x1$

**onedmeshtest4** ()

Yksidimensiainen verkon tasapainotustesti  $mu'' + cu' + ku' = F$

$x0|—x2——|x1$

**onedmeshtest5** ()

Yksidimensiainen verkon tasapainotustesti  $mu'' + cu' + ku' = F$

$x0|——x1——|x2$

**onedmeshtest6** ()

Yksidimensiainen verkon tasapainotustesti  $mu'' + cu' + ku' = F$

$x0|——x1——|x2$

**onedmeshtest7** ()

Kaksidimensioiden verkon tasapainotustesti  $\mu'' + cu' + ku' = f$

kuva prujussa.

**trimeshtest** ()

Kolmioverkon testausta

Created on Tue Mar 20 11:10:29 2012

Laattafunktioita

@author: Jukka

**class PlateCharacteristic** (\*\*kws)

Bases: object

**platechar**

alias of PlateCharacteristic

Created on Sun Mar 11 12:04:59 2012

@author: Jukka

**class Model** (\*args, \*\*kws)

Bases: object

General mechanical model

**assemble\_force\_vector** ()

Assemble force vector

**assemble\_stiffness\_matrix** ()

Assemble stiffness matrix

**assign\_bc** (\*args, \*\*kwargs)

Assigns boundary conditions to elements / nodes

**create\_element\_groups** ()

**create\_node\_groups** ()

**init** ()

**load\_elements**

Decorator that caches a function's return value each time it is called. If called later with the same arguments, the cached value is returned, and not re-evaluated.

**static\_solve** (\*\*kws)

**stiff\_elements**

Decorator that caches a function's return value each time it is called. If called later with the same arguments, the cached value is returned, and not re-evaluated.

**assign\_bc** (\*\*kws)

**assign\_char** (\*\*kws)

**assign\_material** (\*\*kws)

2d sauvamalli

kesken.

**class Bar2D** (\*\*kws)

Bases: object

Sauvaelementti

**B** (\*ke)

Returns plane kinematic matrix B

**assign\_char** (char)

```
assign_material(mat)
```

```
detJ
```

```
stiffness_matrix
```

```
class Bar2DModel(*args, **kws)
```

```
Bases: teefem.models.common.Model
```

```
Bar 2d model
```

```
ex1()
```

```
Ristikkorakenteen ratkaisu
```

```
class C_PLAN(*args, **kws)
```

```
Bases: teefem.models.common.Model
```

```
Plane stress modelisation
```

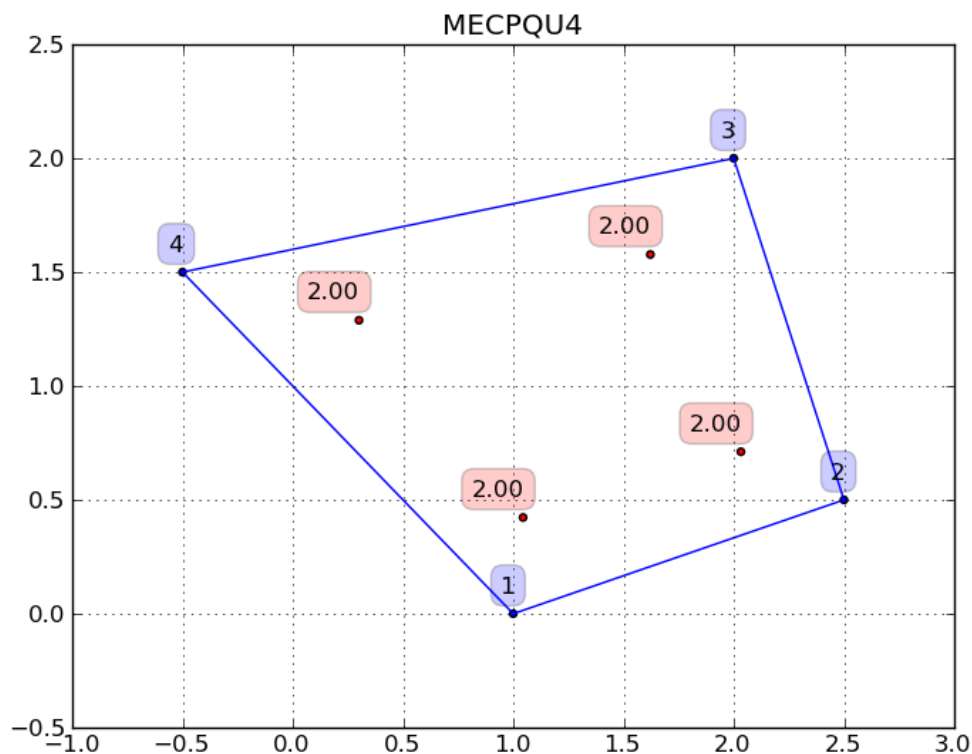
```
class MECPQU4(*args, **kws)
```

```
Bases: teefem.models.c_plan.PlaneStressElement2D
```

```
Plane stress linear quadrangle finite element
```

```
EXAMPLES:
```

```
>>> n1 = Node(x=1.0, y=0.0)
>>> n2 = Node(x=2.5, y=0.5)
>>> n3 = Node(x=2.0, y=2.0)
>>> n4 = Node(x=-0.5, y=1.5)
>>> qu4 = Quad4(name = 'MA1', nodes = (n1, n2, n3, n4))
>>> qu4.material = Elastic(E = 100e9, nu = 0.5)
>>> mal = MECPQU4(shape = qu4)
>>> mal.plot(filename = 'img/MECPQU4.png')
```





```

>>> malemat = mal.material_matrix
>>> print malemat
[[ 1.3e+11  6.7e+10  0.0e+00]
 [ 6.7e+10  1.3e+11  0.0e+00]
 [ 0.0e+00  0.0e+00  3.3e+10]]

>>> malbmat = mal.kinematic_matrix(0,0)
>>> print malbmat
[[-0.1  0.  0.3  0.  0.1  0. -0.3  0. ]
 [ 0. -0.4  0. -0.1  0.  0.4  0.  0.1]
 [-0.4 -0.1 -0.1  0.3  0.4  0.1  0.1 -0.3]]

>>> malkmat = mal.stiffness_matrix
>>> print malkmat
[[ 4.7e+10  2.3e+10 -3.0e+10 -1.1e+10 -1.9e+10 -2.0e+10  2.0e+09  8.6e+09]
 [ 2.3e+10  1.1e+11 -2.8e+10 -2.7e+09 -2.0e+10 -7.0e+10  2.5e+10 -3.9e+10]
 [-3.0e+10 -2.8e+10  6.1e+10 -1.2e+10 -1.5e+09  2.5e+10 -2.9e+10  1.5e+10]
 [-1.1e+10 -2.7e+09 -1.2e+10  4.9e+10  8.3e+09 -4.4e+10  1.5e+10 -2.5e+09]
 [-1.9e+10 -2.0e+10 -1.5e+09  8.3e+09  4.0e+10  2.2e+10 -1.9e+10 -1.0e+10]
 [-2.0e+10 -7.0e+10  2.5e+10 -4.4e+10  2.2e+10  1.0e+11 -2.7e+10  1.3e+10]
 [ 2.0e+09  2.5e+10 -2.9e+10  1.5e+10 -1.9e+10 -2.7e+10  4.7e+10 -1.4e+10]
 [ 8.6e+09 -3.9e+10  1.5e+10 -2.5e+09 -1.0e+10  1.3e+10 -1.4e+10  2.8e+10]]

```

```
class MECPU8 (*args, **kws)
```

Bases: teefem.models.c\_plan.PlaneStressElement2D

Plane stress quadratic 8-node Serendip finite element

```
class MECPU9 (*args, **kws)
```

Bases: teefem.models.c\_plan.PlaneStressElement2D

Plane stress quadratic 9-node finite element

```
class MECPTR3 (*args, **kws)
```

Bases: teefem.models.c\_plan.PlaneStressElement2D

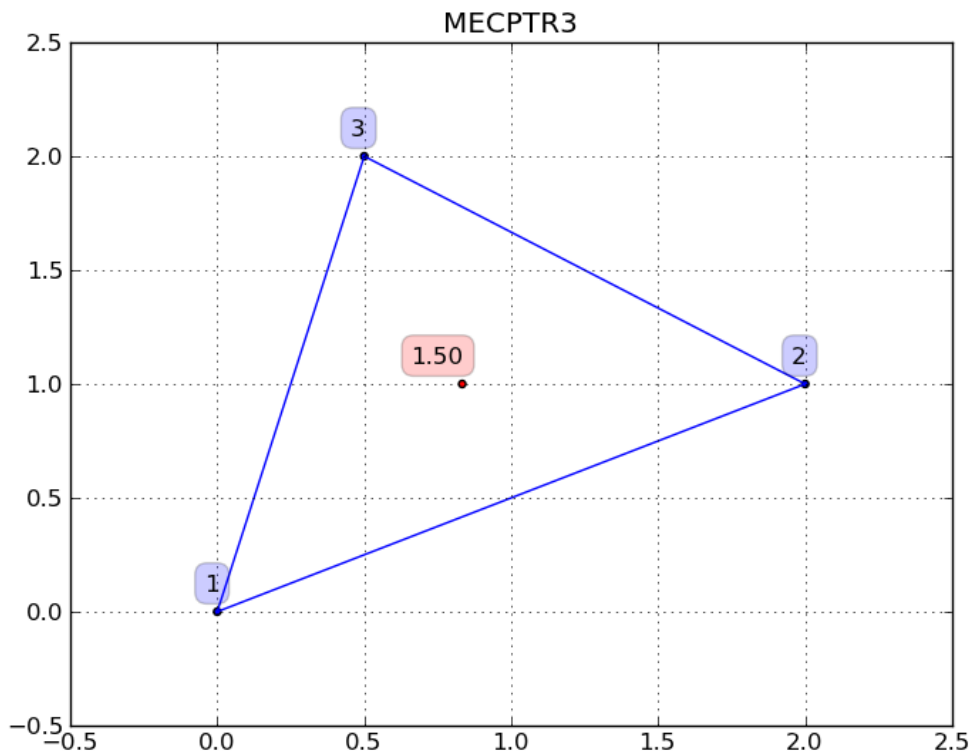
Plane stress linear triangle finite element CST

EXAMPLES:

```

>>> n1 = Node(x=0.0, y=0.0)
>>> n2 = Node(x=2.0, y=1.0)
>>> n3 = Node(x=0.5, y=2.0)
>>> tr3 = Tria3(name = 'tr3', nodes = (n1, n2, n3))
>>> tr3.material = Elastic(E = 210e9, nu = 0.3)
>>> el3 = MECPTR3(shape = tr3)
>>> el3.plot(filename = 'img/MECPTR3.png')

```



```
>>> el3emat = el3.material_matrix
>>> print el3emat
[[ 2.3e+11  6.9e+10  0.0e+00]
 [ 6.9e+10  2.3e+11  0.0e+00]
 [ 0.0e+00  0.0e+00  8.1e+10]]

>>> el3bmat = el3.kinematic_matrix(1/3,1/3)
>>> print el3bmat
[[-0.3  0.   0.6  0.  -0.3  0. ]
 [ 0.  -0.4  0.  -0.1  0.   0.6]
 [-0.4 -0.3 -0.1  0.6  0.6 -0.3]]

>>> el3kmat = el3.stiffness_matrix
>>> print el3kmat
[[ 5.9e+10  3.2e+10 -5.7e+10 -3.0e+10 -1.6e+09 -2.5e+09]
 [ 3.2e+10  8.6e+10 -2.4e+10  1.6e+09 -8.2e+09 -8.7e+10]
 [-5.7e+10 -2.4e+10  1.3e+11 -2.1e+10 -7.7e+10  4.5e+10]
 [-3.0e+10  1.6e+09 -2.1e+10  5.4e+10  5.1e+10 -5.6e+10]
 [-1.6e+09 -8.2e+09 -7.7e+10  5.1e+10  7.9e+10 -4.3e+10]
 [-2.5e+09 -8.7e+10  4.5e+10 -5.6e+10 -4.3e+10  1.4e+11]]
```

```
class MECPTR6(*args, **kws)
```

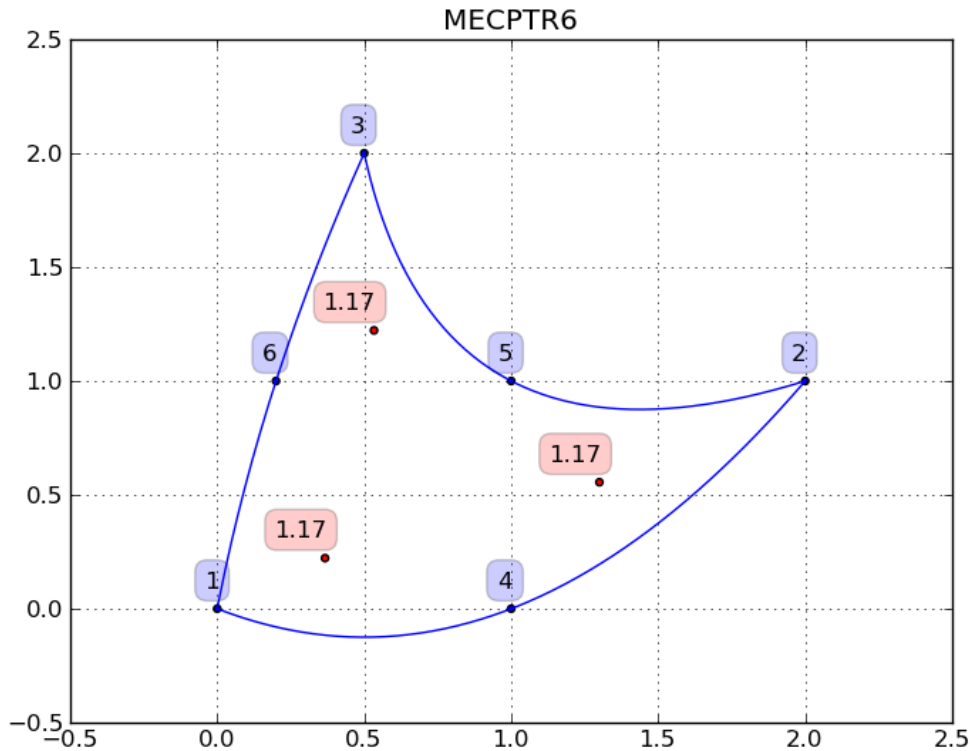
Bases: `teefem.models.c_plan.PlaneStressElement2D`

Plane stress quadratic triangular finite element

EXAMPLES:

```
>>> n1 = Node(x=0.0, y=0.0)
>>> n2 = Node(x=2.0, y=1.0)
>>> n3 = Node(x=0.5, y=2.0)
>>> n4 = Node(x=1.0, y=0.0)
```

```
>>> n5 = Node(x=1.0, y=1.0)
>>> n6 = Node(x=0.2, y=1.0)
>>> tr6 = Tria6(name = 'tr6', nodes = (n1, n2, n3, n4, n5, n6))
>>> tr6.material = Elastic(E = 210e9, nu = 0.3)
>>> el6 = MECPTR6(shape = tr6)
>>> el6.plot(filename = 'img/MECPTR6.png')
```



```
>>> el6emat = el6.material_matrix
>>> print el6emat
[[ 2.3e+11  6.9e+10  0.0e+00]
 [ 6.9e+10  2.3e+11  0.0e+00]
 [ 0.0e+00  0.0e+00  8.1e+10]]

>>> el6bmat = el6.kinematic_matrix(1/3,1/3)
>>> print el6bmat
[[-0.2  0.  0.2  0.  -0.  0.  0.1  0.  0.7  0.  -0.8  0. ]
 [ 0.  -0.2  0.  -0.  0.  0.2  0.  -0.7  0.  0.6  0.  0.1]
 [-0.2 -0.2 -0.  0.2  0.2 -0.  -0.7  0.1  0.6  0.7  0.1 -0.8]]

>>> el6kmat = el6.stiffness_matrix
>>> print el6kmat
[[ 1.9e+11  7.2e+10  2.3e+10  1.3e+10  1.9e+10  1.7e+10 -1.5e+11 -2.0e+10 -9.5e+09
  7.2e+10  1.4e+11  1.1e+10  8.4e+09  1.9e+10  2.7e+10 -1.2e+10 -4.4e+10 -2.5e+10
  2.3e+10  1.1e+10  1.2e+11  4.6e+09  1.5e+10 -1.0e+10 -1.5e+10 -4.4e+10 -1.5e+11
  1.3e+10  8.4e+09  4.6e+09  4.2e+10 -1.2e+10  7.1e+09 -5.1e+10 -2.1e+10  5.6e+10
  1.9e+10  1.9e+10  1.5e+10 -1.2e+10  3.6e+10  9.3e+09  9.9e+08 -2.2e+10  5.3e+09
  1.7e+10  2.7e+10 -1.0e+10  7.1e+09  9.3e+09  8.5e+10 -2.2e+10  1.1e+10  6.2e+10
 -1.5e+11 -1.2e+10 -1.5e+10 -5.1e+10  9.9e+08 -2.2e+10  2.6e+11 -5.5e+10 -6.6e+10
 -2.0e+10 -4.4e+10 -4.4e+10 -2.1e+10 -2.2e+10  1.1e+10 -5.5e+10  3.6e+11  4.1e+10
 -9.5e+09 -2.5e+10 -1.5e+11  5.6e+10  5.3e+09  6.2e+10 -6.6e+10  4.1e+10  5.9e+11
 -2.5e+10 -1.2e+09  4.8e+10 -4.1e+10  7.0e+10 -3.6e+10  4.1e+10 -2.7e+11 -9.6e+10]
```

```
[ -7.4e+10  -6.4e+10   1.2e+10  -9.7e+09  -7.5e+10  -5.7e+10  -2.8e+10   9.9e+10  -3.7e+11
[ -5.7e+10  -1.3e+11  -9.7e+09   3.7e+09  -6.5e+10  -9.5e+10   9.9e+10  -4.5e+10  -3.8e+10
```

**class MEPLSE2** (\*args, \*\*kws)

Bases: `teefem.models.c_plan.PlaneStressElement1D`

Plane stress linear line element.

**class MEPLSE3** (\*args, \*\*kws)

Bases: `teefem.models.c_plan.PlaneStressElement1D`

Plane stress quadratic line element.

**class PlaneStressElement1D** (\*args, \*\*kws)

Bases: `teefem.elements.Element1D`

Plane stress 1d elements, general class.

**assign\_boundary\_condition** (bc)

**force\_vector**

**class PlaneStressElement2D** (\*args, \*\*kws)

Bases: `teefem.elements.Element2D`

Plane stress 2d elements, general class.

**assign\_material** (mat)

**detJ**

Decorator that caches a function's return value each time it is called. If called later with the same arguments, the cached value is returned, and not re-evaluated.

**stiffness\_matrix**

Stiffness matrix K

**update** (U)

**cplan**

alias of `C_PLAN`

## 1.3 Discrete Kirchhoff elements

### 1.3.1 Examples

- `examples/dkt_ex1.py` – `dkt_ex1.py`

**class DKT** (\*args, \*\*kws)

Bases: `teefem.models.common.Model`

Discrete Kirchhoff Triangle model.

**plot** (\*\*kws)

Plottaus.

**class MEBODKT** (\*args, \*\*kws)

Bases: `teefem.elements.Element1D`

SEG2 element in DKT formulation

**class MEDKTR3** (\*args, \*\*kws)

Bases: `teefem.elements.Element2D`

DKT finite element

**U**

**assign\_boundary\_condition** (*bc*)

**assign\_char** (*char*)

**assign\_material** (*mat*)

**detJ**

Decorator that caches a function's return value each time it is called. If called later with the same arguments, the cached value is returned, and not re-evaluated.

**force\_vector**

Palauttaa kuormitusvektorin

**plot** (*\*\*kws*)

Plot geometry with mplot3d

**plot2** (*\*\*kws*)

Plot geometry with mplot3d

**stiffness\_matrix**

Stiffness matrix

**update** (*U*)

**w** (*k, e*)

Jostakin löytyi tämmöset muofofunktiot joilla voisi yrittää interpoloida siirtymäkenttää elementin alueella.

**checkmatrix** (*A*)

**dkt**

alias of DKT

**ex1** ()

Suorakaidelaatta, tasan jakautunut kuorma

**ex2** ()

DKT esimerkki 1, pyörähdysymmetrinen laatta jakautuneella kuormalla

**plotmdl** (*mdl, vmis*)

Plottailee kaikenlaisia kuvaajia

**rotsy** ()

Ympyrälaatan tarkkoja ratkaisuja

Created on Sun Mar 11 02:04:59 2012

@author: Jukka

Mindlin elementti, perusversio.

**class DSTS6TR3** (*\*args, \*\*kws*)

Bases: `teefem.models.mindlin.MINTR3`

DSTS6TR3 TRIA3 Reissner-Mindlin finite element

**Bb** (*\*ke*)

Kinemaattinen matriisi Bb

**Bs** (*\*ke*)

Kinemaattinen matriisi Bs

**force\_vector**

Palauttaa kuormitusvektorin

**class MINDSTS6** (*\*args, \*\*kws*)

Bases: `teefem.models.common.Model`

DST-S6

**plot** (*\*\*kws*)  
Plottaa siirtymäkentän matplotlibillä

**dsts6**  
alias of MINDSTS6

**ex1** ()  
Mindlin esimerkki 1, pyörähdysymmetrinen laatta jakautuneella kuormalla

**shapef** ()  
Pyörittelee DST-6 funktiot

**shapeftest** ()

**test** ()  
DSTS6TR3-elementin matriisien tarkistus

Created on Sun Mar 11 02:04:59 2012

@author: Jukka

Mindlin elementti. Tessler ja Hughesin versio.

**class MINTH** (*\*args, \*\*kws*)  
Bases: `teefem.models.mindlin.MindlinModel`  
Reissner-Mindlin Tessler & Hughes model

**class MINTR3TH** (*\*args, \*\*kws*)  
Bases: `teefem.models.mindlin.MINTR3`  
Mindlin TRIA3 Tessler & Hughes

**Bs** (*\*ke*)  
Kinemaattinen matriisi Bs

**ex1** ()  
Mindlin esimerkki 1, pyörähdysymmetrinen laatta jakautuneella kuormalla

**genshapef** ()  
Tessler ja Hughes muotofunktioiden pyörittelyä

**minth**  
alias of MINTH

Created on Sun Mar 11 12:04:59 2012

@author: Jukka

Mindlin elementti, perusversio.

**class MIN** (*\*args, \*\*kws*)  
Bases: `teefem.models.mindlin.MindlinModel`  
Reissner-Mindlin model

**class MINR** (*\*args, \*\*kws*)  
Bases: `teefem.models.mindlin.MindlinModel`  
Reissner-Mindlin model, reduced integration

**class MINTR3** (*\*args, \*\*kws*)  
Bases: `teefem.elements.Element2D`  
MINTR3 TRIA3 Reissner-Mindlin finite element

**Bb** (*\*ke*)  
Kinemaattinen matriisi Bb

**Bs** (*\*ke*)  
Kinemaattinen matriisi Bs

**Db** (*\*ke*)

**Ds** (*\*ke*)

**assign\_boundary\_condition** (*bc*)

**assign\_char** (*char*)

**assign\_material** (*mat*)

**force\_vector**  
Palauttaa kuormitusvektorin

**stiffness\_matrix**  
Jäykkyysmatriisi

**update** (*U*)

**class MINTR3R** (*\*args, \*\*kws*)  
Bases: `teefem.models.mindlin.MINTR3`  
MINTR3 TRIA3 Reissner-Mindlin finite element, reduced integration

**class MindlinModel** (*\*args, \*\*kws*)  
Bases: `teefem.models.common.Model`  
Mindlin model

**plot** (*\*\*kws*)  
Plottaa siirtymäkentän matplotlibillä

**ex1** ()  
Mindlin esimerkki 1, pyörähdysymmetrinen laatta jakautuneella kuormalla

**mindlin**  
alias of MIN

**mindlin\_reduced**  
alias of MINR

**test** ()  
MINTR3SI-elementin matriisien tarkistus





# MODULE INDEX

## T

- teefem, 3
- teefem.boundary\_conditions, 6
- teefem.cache, 7
- teefem.common, 7
- teefem.deprecation, 8
- teefem.elements, 8
- teefem.geom, 3
- teefem.helpers, 8
- teefem.materials, 8
- teefem.mesh, 9
- teefem.meshtests, 10
- teefem.models.bar2d, 11
- teefem.models.c\_plan, 12
- teefem.models.common, 11
- teefem.models.dkt, 16
- teefem.models.dsts6, 17
- teefem.models.min3, 18
- teefem.models.mindlin, 18
- teefem.plate\_functions, 11



# INDEX

## A

add() (teefem.common.IJV method), 8  
 anglxy() (in module teefem.common), 8  
 arctan2() (in module teefem.meshtests), 10  
 area (teefem.elements.Element attribute), 8  
 arrtostr() (in module teefem.common), 8  
 assemble\_force\_vector()  
     (teefem.models.common.Model method), 11  
 assemble\_stiffness\_matrix()  
     (teefem.models.common.Model method), 11  
 assign\_bc() (in module teefem.models.common), 11  
 assign\_bc() (teefem.models.common.Model method),  
     11  
 assign\_boundary\_condition() (teefem.geom.Node  
     method), 4  
 assign\_boundary\_condition()  
     (teefem.models.c\_plan.PlaneStressElement1D  
     method), 16  
 assign\_boundary\_condition()  
     (teefem.models.dkt.MEDKTR3 method),  
     16  
 assign\_boundary\_condition()  
     (teefem.models.mindlin.MINTR3 method),  
     19  
 assign\_char() (in module teefem.models.common), 11  
 assign\_char() (teefem.models.bar2d.Bar2D method),  
     11  
 assign\_char() (teefem.models.dkt.MEDKTR3 method),  
     17  
 assign\_char() (teefem.models.mindlin.MINTR3  
     method), 19  
 assign\_material() (in module teefem.models.common),  
     11  
 assign\_material() (teefem.geom.Mesh method), 3  
 assign\_material() (teefem.models.bar2d.Bar2D  
     method), 12  
 assign\_material() (teefem.models.c\_plan.PlaneStressElement1D  
     method), 16  
 assign\_material() (teefem.models.dkt.MEDKTR3  
     method), 17  
 assign\_material() (teefem.models.mindlin.MINTR3  
     method), 19  
 average\_edge\_lenght (teefem.mesh.Trimesh attribute),  
     9

## B

B() (teefem.models.bar2d.Bar2D method), 11  
 Bar2D (class in teefem.models.bar2d), 11  
 Bar2DModel (class in teefem.models.bar2d), 12  
 Bb() (teefem.models.dsts6.DSTS6TR3 method), 17  
 Bb() (teefem.models.mindlin.MINTR3 method), 18  
 BoundaryCondition (class in  
     teefem.boundary\_conditions), 6  
 Bs() (teefem.models.dsts6.DSTS6TR3 method), 17  
 Bs() (teefem.models.min3.MINTR3TH method), 18  
 Bs() (teefem.models.mindlin.MINTR3 method), 18

## C

C\_PLAN (class in teefem.models.c\_plan), 12  
 Cache (in module teefem.cache), 7  
 cache (in module teefem.cache), 7  
 checkmatrix() (in module teefem.models.dkt), 17  
 cplan (in module teefem.models.c\_plan), 16  
 create\_element\_groups()  
     (teefem.models.common.Model method), 11  
 create\_node\_groups() (teefem.geom.Mesh method), 3  
 create\_node\_groups() (teefem.models.common.Model  
     method), 11

## D

Db() (teefem.models.mindlin.MINTR3 method), 18  
 deprecated() (in module teefem.deprecation), 8  
 detJ (teefem.models.bar2d.Bar2D attribute), 12  
 detJ (teefem.models.c\_plan.PlaneStressElement2D at-  
     tribute), 16  
 detJ (teefem.models.dkt.MEDKTR3 attribute), 17  
 detJ() (teefem.geom.Net2D method), 3  
 dirichlet\_bc (in module teefem.boundary\_conditions),  
     7  
 DirichletBoundaryCondition (class in  
     teefem.boundary\_conditions), 6  
 dist() (in module teefem.common), 8  
 dist() (in module teefem.meshtests), 10  
 DKT (class in teefem.models.dkt), 16  
 dkt (in module teefem.models.dkt), 17  
 Ds() (teefem.models.mindlin.MINTR3 method), 19  
 dsts6 (in module teefem.models.dsts6), 18  
 DSTS6TR3 (class in teefem.models.dsts6), 17

## E

Elastic (class in teefem.materials), 9

elastic (in module teefem.materials), 9  
Element (class in teefem.elements), 8  
Element1D (class in teefem.elements), 8  
Element2D (class in teefem.elements), 8  
ex1() (in module teefem.models.bar2d), 12  
ex1() (in module teefem.models.dkt), 17  
ex1() (in module teefem.models.dsts6), 18  
ex1() (in module teefem.models.min3), 18  
ex1() (in module teefem.models.mindlin), 19  
ex2() (in module teefem.models.dkt), 17  
exist() (teefem.boundary\_conditions.DirichletBoundaryCondition method), 7  
exist() (teefem.boundary\_conditions.NodalForceCondition method), 7

## F

fibonacci (in module teefem.cache), 7  
find\_group\_ma() (teefem.geom.Mesh method), 3  
find\_group\_no() (teefem.geom.Mesh method), 3  
force\_eq() (in module teefem.mesh), 9  
force\_eq\_test1() (in module teefem.mesh), 9  
force\_vector (teefem.models.c\_plan.PlaneStressElement1D attribute), 16  
force\_vector (teefem.models.dkt.MEDKTR3 attribute), 17  
force\_vector (teefem.models.dsts6.DSTS6TR3 attribute), 17  
force\_vector (teefem.models.mindlin.MINTR3 attribute), 19  
forcemoc() (in module teefem.mesh), 9  
forcemoc\_test1() (in module teefem.mesh), 9

## G

genshapef() (in module teefem.models.min3), 18

## H

hasbc() (teefem.geom.Node method), 4  
hasload() (teefem.geom.Node method), 4

## I

IJV (class in teefem.common), 7  
init() (teefem.models.common.Model method), 11  
invJ (teefem.elements.Element attribute), 8  
invJ() (teefem.geom.Net2D method), 3

## J

J (teefem.elements.Element attribute), 8  
J() (teefem.geom.Net2D method), 3

## L

load\_elements (teefem.models.common.Model attribute), 11

## M

MEBODKT (class in teefem.models.dkt), 16  
MECPQU4 (class in teefem.models.c\_plan), 12  
MECPQU8 (class in teefem.models.c\_plan), 13

MECPQU9 (class in teefem.models.c\_plan), 13  
MECPTR3 (class in teefem.models.c\_plan), 13  
MECPTR6 (class in teefem.models.c\_plan), 14  
MEDKTR3 (class in teefem.models.dkt), 16  
MEPLSE2 (class in teefem.models.c\_plan), 16  
MEPLSE3 (class in teefem.models.c\_plan), 16  
Mesh (class in teefem.geom), 3  
mesh() (in module teefem.geom), 6  
MIN (class in teefem.models.mindlin), 18  
mindlin (in module teefem.models.mindlin), 19  
mindlin\_reduced (in module teefem.models.mindlin), 19  
MindlinModel (class in teefem.models.mindlin), 19  
MINDSTS6 (class in teefem.models.dsts6), 17  
MINR (class in teefem.models.mindlin), 18  
MINTH (class in teefem.models.min3), 18  
minth (in module teefem.models.min3), 18  
MINTR3 (class in teefem.models.mindlin), 18  
MINTR3R (class in teefem.models.mindlin), 19  
MINTR3TH (class in teefem.models.min3), 18  
Model (class in teefem.models.common), 11

## N

Net (class in teefem.geom), 3  
Net1D (class in teefem.geom), 3  
Net2D (class in teefem.geom), 3  
nodal\_force (in module teefem.boundary\_conditions), 7  
NodalForce (class in teefem.boundary\_conditions), 7  
Node (class in teefem.geom), 3

## O

onedmeshtest() (in module teefem.meshtests), 10  
onedmeshtest2() (in module teefem.meshtests), 10  
onedmeshtest3() (in module teefem.meshtests), 10  
onedmeshtest4() (in module teefem.meshtests), 10  
onedmeshtest5() (in module teefem.meshtests), 10  
onedmeshtest6() (in module teefem.meshtests), 10  
onedmeshtest7() (in module teefem.meshtests), 10

## P

parsemail() (in module teefem.geom), 6  
parsemed() (in module teefem.geom), 6  
parsemesh() (in module teefem.geom), 6  
PlaneStressElement1D (class in teefem.models.c\_plan), 16  
PlaneStressElement2D (class in teefem.models.c\_plan), 16  
platechar (in module teefem.plate\_functions), 11  
PlateCharacteristic (class in teefem.plate\_functions), 11  
plot() (teefem.elements.Element2D method), 8  
plot() (teefem.geom.Net2D method), 3  
plot() (teefem.geom.Quad8 method), 5  
plot() (teefem.mesh.Trimesh method), 9  
plot() (teefem.models.dkt.DKT method), 16  
plot() (teefem.models.dkt.MEDKTR3 method), 17  
plot() (teefem.models.dsts6.MINDSTS6 method), 17

plot() (teefem.models.mindlin.MindlinModel method), 19  
 plot2() (teefem.models.dkt.MEDKTR3 method), 17  
 plot3d() (teefem.geom.Net2D method), 3  
 plotmdl() (in module teefem.models.dkt), 17  
 Poi1 (class in teefem.geom), 4  
 pressure\_bc (in module teefem.boundary\_conditions), 7  
 PressureBoundaryCondition (class in teefem.boundary\_conditions), 7

## Q

Quad4 (class in teefem.geom), 4  
 Quad8 (class in teefem.geom), 5  
 Quad9 (class in teefem.geom), 5

## R

refine() (in module teefem.mesh), 9  
 rotsy() (in module teefem.models.dkt), 17

## S

Seg2 (class in teefem.geom), 5  
 Seg3 (class in teefem.geom), 5  
 sgm() (teefem.mesh.Trimesh method), 9  
 sgmtest() (in module teefem.mesh), 9  
 shapef() (in module teefem.models.dsts6), 18  
 shapeftest() (in module teefem.models.dsts6), 18  
 static\_solve() (teefem.models.common.Model method), 11  
 status (teefem.elements.Element attribute), 8  
 status() (teefem.geom.Mesh method), 3  
 status() (teefem.geom.Node method), 4  
 stiff\_elements (teefem.models.common.Model attribute), 11  
 stiffness\_matrix (teefem.models.bar2d.Bar2D attribute), 12  
 stiffness\_matrix (teefem.models.c\_plan.PlaneStressElement2D attribute), 16  
 stiffness\_matrix (teefem.models.dkt.MEDKTR3 attribute), 17  
 stiffness\_matrix (teefem.models.mindlin.MINTR3 attribute), 19

## T

teefem (module), 3  
 teefem.boundary\_conditions (module), 6  
 teefem.cache (module), 7  
 teefem.common (module), 7  
 teefem.deprecation (module), 8  
 teefem.elements (module), 8  
 teefem.geom (module), 3  
 teefem.helpers (module), 8  
 teefem.materials (module), 8  
 teefem.mesh (module), 9  
 teefem.meshtests (module), 10  
 teefem.models.bar2d (module), 11  
 teefem.models.c\_plan (module), 12  
 teefem.models.common (module), 11

teefem.models.dkt (module), 16  
 teefem.models.dsts6 (module), 17  
 teefem.models.min3 (module), 18  
 teefem.models.mindlin (module), 18  
 teefem.plate\_functions (module), 11  
 test() (in module teefem.models.dsts6), 18  
 test() (in module teefem.models.mindlin), 19  
 test1() (in module teefem.boundary\_conditions), 7  
 test1() (in module teefem.common), 8  
 test1() (in module teefem.mesh), 9  
 tocoo() (teefem.common.IJV method), 8  
 Tria3 (class in teefem.geom), 5  
 Tria6 (class in teefem.geom), 6  
 Trimesh (class in teefem.mesh), 9  
 trimeshtest() (in module teefem.meshtests), 11

## U

U (teefem.models.dkt.MEDKTR3 attribute), 16  
 unitcircle() (in module teefem.mesh), 10  
 unitcircle1() (in module teefem.mesh), 10  
 unitcircle2() (in module teefem.mesh), 10  
 unitinterval() (in module teefem.mesh), 10  
 unitsquare() (in module teefem.mesh), 10  
 unitsquare2() (in module teefem.mesh), 10  
 update() (teefem.elements.Element method), 8  
 update() (teefem.models.c\_plan.PlaneStressElement2D method), 16  
 update() (teefem.models.dkt.MEDKTR3 method), 17  
 update() (teefem.models.mindlin.MINTR3 method), 19  
 update\_boundary\_conditions() (teefem.geom.Node method), 4  
 update\_field() (teefem.geom.Node method), 4

## V

val() (teefem.boundary\_conditions.DirichletBoundaryCondition method), 7  
 val() (teefem.boundary\_conditions.NodalForce method), 7  
 valbc() (teefem.geom.Node method), 4  
 valload() (teefem.geom.Node method), 4

## W

w() (teefem.models.dkt.MEDKTR3 method), 17