Classical Laminate Theory Calculator

Release 0.0.9

A.J.J. Lagerweij

Apr 17, 2023

Table of Contents

CHAPTER 1

Introduction

This set of scrips contains functions that can be used for the design of simple fibre composite laminates. The goal of this project is to create a open-source laminate theory calculator something that can compete with "The Laminator". There are two benefits of the project in python is that it can easily be integrated in, for example, a stiffend plate buckling/cripling calculator. Or that it can be used to find the optimal layout by iterating over various designs.

Before using the code I recommend to read upon literature introducing the Classical Laminate Theory. Many such books exist, from my experience the following where reccomended:

- Design and Analysis of Composite Structures: With Applications to Aerospace Structures by Christos Kassapoglou (ISBN: 9781118536933, DOI: 10.1002/9781118536933)
- Composite Materials: Design and Applications by Daniel Gay (ISBN: 9780429101038 DOI: 10.1201/b17106)

Todo:

- Adding functions to the homonigization module.
- Adding a inverse ply failure calculator for Tsai-Hill.
- Adding buckling / cripling calculators.

1.1 Ply Homogenization Calculator

1.1.1 Introduction

This file containts functions to calculate the ply properties based upon the the properties of the constituents.

Note: Currently it does only convert mass fraction into volume fractions and the other way around. In the future a Rule of Mixture and a Tsai honogenization function will be added.

Table of Contents

1.1.2 Routines

Calculate the ply properties from the properties of the constituents.

A.J.J. Lagerweij COHMAS Mechanical Engineering KAUST 2020

`homogenization.`**`isotropic2D`**(*E*, *nu*)

Determine stiffness & compliance matrix of 2D isotropic material.

Parameters

- **E** (*float*) – Young's modulus.
- **nu** (*float*) – Poisson's ratio.

Returns

- **C** (*matrix*) – 3D stiffness matrix in Voigt notation (6x6).
- **S** (*matrix*) – 3D compliance matrix in Voigt notation (6x6).

`homogenization.`**`isotropic3D`**(*E*, *nu*)

Determine stiffness & compliance matrix of 3D isotropic material.

Parameters

- **E** (*float*) – Young's modulus.
- **nu** (*float*) – Poisson's ratio.

Returns

- **C** (*matrix*) – 3D stiffness matrix in Voigt notation (6x6).
- **S** (*matrix*) – 3D compliance matrix in Voigt notation (6x6).

`homogenization.`**`massfrac_to_volfrac`**(*fm1*, *rho1*, *rho2*)

Caluculate the volume fraction from the mass fraction.

Parameters

- **fm1** (*float*) – The mass fraction of material 1 defined as, fm1 = massa 1 / massa total.
- **rho1** (*float*) – The density of material 1.
- **rho2** (*float*) – The density of material 2.

Returns

- **fv1** (*float*) – The fraction material 1 is used in volume.
- **fv2** (*float*) – The fraction material 2 is used in volume.

`homogenization.`**`orthotropic3D`**(*E1*, *E2*, *E3*, *nu12*, *nu13*, *nu23*, *G12*, *G13*, *G23*)

Determine stiffness & compliance matrix of a 3D orthotropic material.

This notation is in Voigt notation with engineering strain $\gamma_{12} = 2\varepsilon_{12}$.

Parameters

- **`E1`** (*`float`*) – Young's modulus in 1 direction.
- **`E2`** (*`float`*) – Young's modulus in 2 direction.
- **`E3`** (*`float`*) – Young's modulus in 3 direction.
- **`nu12`** (*`float`*) – Poisson's ratio over 12.
- **`nu13`** (*`float`*) – Poisson's ratio over 13.
- **`nu23`** (*`float`*) – Poisson's ratio over 23.
- **`G12`** (*`float`*) – Shear modulus over 12.
- **`G13`** (*`float`*) – Shear modulus over 13.
- **`G23`** (*`float`*) – Shear modulus over 23.

Returns

- **C** (*matrix*) – 2D stiffness matrix in Voigt notaion (6x6).
- **S** (*matrix*) – 2D compliance matrix in Voigt notation (6x6).

`homogenization.`**`reuss`**(*S1*, *S2*, *volf*)

Calculate the compliance matrix with the Reuss limit.

This lower limit of the rule of mixtures is generaly used for the transverse stiffness E_l.

Parameters

- **`S1`** (*`matrix`*) – The compliance matrix of material 1.
- **`S2`** (*`float`*) – The compliance matrix of material 2.
- **`volf`** (*`matrix`*) – The volume fraction of material 1.

Returns

- **C_hat** (*matrix*) – The stiffess matrix of the mixed material.
- **S_hat** (*matrix*) – The compliance matrix of the mixed material.

`homogenization.`**`trans_isotropic2D`**(*E1*, *E2*, *nu12*, *G12*)

Determine stiffness & compliance matrix of 2D plane stress transverse isotropic material.

This notation is in Voigt notation with engineering strain $\gamma_{12} = 2\varepsilon_{12}$.

Parameters

- **`E1`** (*`float`*) – Young's modulus in 1 direction.
- **`E2`** (*`float`*) – Young's modulus in 3 direction.
- **`nu12`** (*`float`*) – Poisson's ratio over 12.
- **`G12`** (*`float`*) – Shear modulus over 13.

Returns

- **C** (*matrix*) – 2D stiffness matrix in Voigt notaion (3x3).
- **S** (*matrix*) – 2D compliance matrix in Voigt notation (3x3).

homogenization.**trans_isotropic3D**(*E1*, *E2*, *nu12*, *nu23*, *G12*)

Determine stiffness & compliance matrix of 3D transverse isotropic material.

This notation is in Voigt notation with engineering strain $\gamma_{12} = 2\varepsilon_{12}$.

Parameters

- **E1** (*float*) – Young's modulus in 1 direction.
- **E3** (*float*) – Young's modulus in 3 direction.
- **nu12** (*float*) – Poisson's ratio over 12.
- **nu23** (*float*) – Poisson's ratio over 23.
- **G13** (*float*) – Shear modulus over 13.

Returns

- **C** (*matrix*) – 2D stiffness matrix in Voigt notaion (6x6).
- **S** (*matrix*) – 2D compliance matrix in Voigt notation (6x6).

homogenization.**voigt**(*C1*, *C2*, *volf*)

Calculate the mixed stiffness matrix with the Voigt.

This upper limit of the rule of mixtures is generaly used for the longitudional stiffness E_l.

Parameters

- **C1** (*matrix*) – The stiffness matrix of material 1.
- **C2** (*float*) – The stiffness matrix of material 2.
- **volf** (*matrix*) – The volume fraction of material 1.

Returns

- **C_hat** (*matrix*) – The stiffness matrix of the mixed material.
- **S_hat** (*matrix*) – The compliance matrix of the mixed material.

homogenization.**volfrac_to_massfrac**(*fv1*, *rho1*, *rho2*)

Caluculate the mass fraction from the volume fraction.

Parameters

- **fv1** (*float*) – The volume fraction of material 1 defined as, fv1 = volume 1 / volume total.
- **rho1** (*float*) – The density of material 1.
- **rho2** (*float*) – The density of material 2.

Returns

- **fm1** (*float*) – The fraction material 1 is used in mass
- **fm2** (*float*) – The fraction material 2 is used in mass.

1.1.3 Indices and Tables

- genindex
- modindex
- search

1.2 ABD Matrix Calculator

1.2.1 Introduction

This file contains functions to analyze the properties of a laminate. It will allow the calculation of the stiffness and compliance matrix for plane stress and it can execute a given stacking sequence to calculate the ADB matrix.

Table of Contents

1.2.2 Routines

ABD matrix calculator for a given stacking sequence.

The methods in this file will call create a ABD matrix of a composit for given ply properties and stacking sequence.

A.J.J. Lagerweij COHMAS Mechanical Engineering KAUST 2020

`abdcal.`**`QPlaneStrain`**(*El*, *Et*, *nult*, *G*)

Generate the plane strain local stiffness matrix.

> **Warning:** Not yet implemented. It raises a *NotImplementedError*.

Parameters

- **El** (*float*) – Elastic modulus in the longitudinal direction.
- **Et** (*float*) – Elastic modulus in the transverse direction.
- **nult** (*float*) – Poisson ratio in longitudinal-transverse direction.
- **G** (*float*) – Shear modulus in longitudinal-transverse directions.

Returns **Q** – Stiffness matrix in longitudinal-transverse directions.

Return type matrix

`abdcal.`**`QPlaneStress`**(*El*, *Et*, *nult*, *G*)

Generate the plane stress local stiffness matrix.

Parameters

- **El** (*float*) – Elastic modulus in the longitudinal direction.
- **Et** (*float*) – Elastic modulus in the transverse direction.
- **nult** (*float*) – Poisson ratio in longitudinal-transverse direction.
- **G** (*float*) – Shear modulus in longitudinal-transverse directions.

Returns **Q** – Stiffness matrix in longitudinal-transverse directions.

Return type matrix

abdcal.**abd**(*Q*, *angles*, *thickness*, *truncate=False*)

Calculate the full ABD matrix of a laminate.

Top plies should be listed first in the lists of Q, angles and thickness.

Parameters

- **Q** (*list*) – The stiffness matrix of each ply in its l-t axis system.
- **angles** (*list*) – The rotation of each ply in degrees.
- **thickness** (*list*) – The thickness of each ply.
- **truncate** (*bool*) – Truncates very small numbers when true.

Returns **ABD** – The stiffness matrix of the thin laminate.

Return type matrix

abdcal.**abdthin**(*Q*, *angles*, *thickness*, *truncate=False*)

ABD matrix calculator for a thin laminate.

In the thin laminate theroy it is assumed that the out of plane stiffness is negligible and that the layup is symmetric. Hence only the membrane (A part) of the ABD matrix remains. Top plies should be listed first in the lists of Q, angles and thickness.

Parameters

- **Q** (*list*) – The stiffness matrix of each ply in its l-t axis system.
- **angles** (*list*) – The rotation of each ply in degrees.
- **thickness** (*list*) – The thickness of each ply.
- **truncate** (*bool*) – Truncates very small numbers when true.

Returns **C** – The stiffness matrix of the thin laminate.

Return type matrix

abdcal.**compliance_rotation**(*compliance*, *angle*)

Rotate the compliance matrix over a given angle.

This rotates the complianc matrix from local to the global axis sytem. Use a negative angle to rotate from global to local system.

Parameters

- **compliance** (*matrix*) – The matrix that must be rotated.
- **angle** (*float*) – The rotation angle in degrees.

Returns **stiffness_rot** – A rotated version of the matrix.

Return type matrix

abdcal.**cte**(*Q*, *angles*, *thickness*, *alpha*)

Coefficient of Thermal Expansion calculator.

This funcion calculates the CTE in the x-y axis sytem. It summs up the rotated CTE of each layer and multiplies them by layer stiffness and thickness. The resulting CTE relates thermal change (ΔT) to the deformation vector (strains and curvatures). Top plies should be listed first in the lists of Q, angles, thickness and alpha.

Parameters

- **Q** (*list*) – The stiffness matrix of each ply in its l-t axis system.

- **angles** (*list*) – The rotation of each ply in degrees.
- **thickness** (*list*) – The thickness of each ply.
- **alpha** (*list*) – The coeficcient of termal expansion of each ply in l-t axis system.

Returns **cte** – The coefficient of termal expansion of the laminate, in x-y axis sytem.

Return type vector

abdcal.**ctethin**(*Q*, *angles*, *thickness*, *alpha*)

Thin laminate Coefficient of Thermal Expansion calculator.

This funcion calculates the CTE in the x-y axis sytem. It summs up the rotated CTE of each layer and weights them by layer stiffness and thickness. Here it is assumed that there is no bending behaviour this is only true for thin and symmetric layups. Top plies should be listed first in the lists of Q, angles, thickness and alpha.

Parameters

- **Q** (*list*) – The stiffness matrix of each ply in its l-t axis system.
- **angles** (*list*) – The rotation of each ply in degrees.
- **thickness** (*list*) – The thickness of each ply.
- **alpha** (*list*) – The coeficcient of termal expansion of each ply in l-t axis system.

Returns **cte** – The coefficient of termal expansion of the laminate, in x-y axis sytem.

Return type vector

abdcal.**matrix_inverse**(*matrix*)

Inverts a matrix.

Parameters **matrix** (*matrix*) – The matrix to be inverted.

Returns **inverse** – The inverse of the matrix.

Return type matrix

abdcal.**stiffness_rotation**(*stiffness*, *angle*)

Rotate the stiffness matrix against given angle.

This rotates the stiffness matrix from local to the global axis sytem. Use a negative angle to rotate from global to local system.

Parameters

- **stiffness** (*matrix*) – The matrix that must be rotated.
- **angle** (*float*) – The rotation angle in degrees.

Returns **stiffness_rot** – A rotated version of the matrix.

Return type matrix

1.2.3 Indices and Tables

- genindex
- modindex
- search

1.3 Deformation Calculator

1.3.1 Introduction

This file containts functions to calculate what the deformation or load on a laminate is. To perform this calculation the ABD matrix and its inverse are required in combination with either:

- Load vector $(N_x, N_y, N_{xy}, M_x, M_y, M_{xy})^T$
- Deformation vector $(\varepsilon_x, \varepsilon_y, \varepsilon_{xy}, \kappa_x, \kappa_y, \kappa_{xy})^T$

Eventually it will calculate the stress and strain values inside each ply of the laminate.

Table of Contents

1.3.2 Routines

Calculate the resulting deformation or stresses for loading conditions.

A.J.J. Lagerweij COHMAS Mechanical Engineering KAUST 2020

`deformation.`**`deformation_applied`**(*abd*, *deformation*)

Calculate the running load and moment of the plate under a given using Kichhoff plate theory.

Parameters

- **abd** (*matrix*) – The ABD matrix.
- **deformation** (*vector*) – This deformation consists of $(\varepsilon_x, \varepsilon_y, \varepsilon_{xy}, \kappa_x, \kappa_y, \kappa_{xy})^T$

Returns **load** – The load vector consits of are $(N_x, N_y, N_{xy}, M_x, M_y, M_{xy})^T$

Return type vector

`deformation.`**`load_applied`**(*abd_inv*, *load*)

Calculate the strain and curvature of the full plate under a given load using Kirchhoff plate theory.

Parameters

- **abd** (*matrix*) – The inverse of the ABD matrix.
- **load** (*vector*) – The load vector consits of are $(N_x, N_y, N_{xy}, M_x, M_y, M_{xy})^T$

Returns **deformation** – This deformation consists of $(\varepsilon_x, \varepsilon_y \varepsilon_{xy}, \kappa_x, \kappa_y, \kappa_{xy})^T$

Return type vector

`deformation.`**`ply_strain`**(*deformed*, *Q*, *angles*, *thickness*)

Calculate the strain at the top and bottom of each ply.

Small and linear deformations are assumed. For each ply two strain states are retured, one for the top and bottom of each ply. As bending moments can leed to different stresses depending on the z location in the ply. Top plies should be listed first in the lists of Q, angles and thickness..

Parameters

- **deformed** (*vector*) – This deformation consists of $(\varepsilon_x, \varepsilon_y \varepsilon_{xy}, \kappa_x, \kappa_y, \kappa_{xy})^T$
- **Q** (*list*) – The local stiffness matrix of each ply.
- **angles** (*list*) – The rotation of each ply in degrees.
- **thickness** (*list*) – The thickness of each ply.
- **plotting** (*bool, optional*) – Plotting the stress distribution or not.

Returns **strain** – The stress vector $(\sigma_{xx}, \sigma_{yy}, \tau_{xy})^T$ of the top, middle and bottom of each ply in the laminate.

Return type list

`deformation.`**`ply_stress`**(*deformed*, *Q*, *angles*, *thickness*, *plotting=False*)

Calculate the stresses at the top and bottom of each ply.

Small and linear deformations are assumed. For each ply two stress states are retured, one for the top and bottom of each ply. As bending moments can leed to different stresses depending on the z location in the ply. Top plies should be listed first in the lists of Q, angles and thickness.

If required the stresses can be plotted as a function of the z coordinates. In this plot the stresses shown are in the global axsis system x and y.

Parameters

- **deformed** (*vector*) – This deformation consists of $(\varepsilon_x, \varepsilon_y \varepsilon_{xy}, \kappa_x, \kappa_y, \kappa_{xy})^T$
- **Q** (*list*) – The local stiffness matrix of each ply.
- **angles** (*list*) – The rotation of each ply in degrees.
- **thickness** (*list*) – The thickness of each ply.
- **plotting** (*bool, optional*) – Plotting the through thickness stress distribution or not.

Returns **stress** – The stress vector $(\sigma_{xx}, \sigma_{yy}, \tau_{xy})^T$ of the top, middle and bottom of each ply in the laminate.

Return type list

`deformation.`**`ply_stress_thermal`**(*deformed*, *angles*, *Q*, *thickness*, *alpha*, *dT*)

Calculate the stress due to mechanical and thermal deformation.

Parameters

- **deformed** (*vector*) – This deformation of the entire laminate. $(\varepsilon_x, \varepsilon_y, \varepsilon_{xy}, \kappa_x, \kappa_y, \kappa_{xy})^T$
- **Q** (*list*) – The local stiffness matrix of each ply in l-t axis system.
- **angles** (*list*) – The rotation of each ply in degrees.
- **thickness** (*list*) – The thickness of each ply.
- **alpha** (*list*) – The coeficcient of termal expansion of each ply in l-t axis system.
- **dT** (*float*) – Change in temperature.

Returns **stress** – The stress vector $(\sigma_{xx}, \sigma_{yy}, \tau_{xy})^T$ of the top, middle and bottom of each ply in the laminate.

Return type list

`deformation.`**`strain_rotation`**(*strain*, *angle*)

Rotates a strain vector against a given angle.

This rotates the strain from local to the global axis sytem. Use a negative angle to rotate from global to local system. The strain vector must be in Voigt notation and engineering strain is used.

Parameters

- **`strain`** (*`vector`*) – The matrix that must be rotated.
- **`angle`** (*`float`*) – The rotation angle in degrees.

Returns **strain_rot** – A rotated version of the matrix.

Return type vector

`deformation.`**`stress_rotation`**(*stress*, *angle*)

Rotates a stress vector against a given angle.

This rotates the stress from local to the global axis sytem. Use a negative angle to rotate from global to local system. The stress vector must be in Voigt notation and engineering stress is used.

Parameters

- **`stress`** (*`vector`*) – The matrix that must be rotated.
- **`angle`** (*`float`*) – The rotation angle in degrees.

Returns **stress_rot** – A rotated version of the matrix.

Return type vector

1.3.3 Indices and Tables

- genindex
- modindex
- search

1.4 Ply Failure Calculator

1.4.1 Introduction

This file containts functions to determine the if the loaded laminate fails acording to one of the following failure criteria:

- Max stress
- Tsai-Wu
- Tsai-Hill

Note: An extension is planned for the Puck critria and some 'inverse' method. These inverse methods will return a scaling factor of which the load can be multiplied with before failure is reached.

Table of Contents

1.4.2 Routines

Calculate the failure (critera) of the composite material.

A.J.J. Lagerweij COHMAS Mechanical Engineering KAUST 2020

`failure.`**`max_stress`**(*stress*, *sl_max*, *sl_min*, *st_max*, *st_min*, *tlt_max*)

Compare stresses to the max stress criteria, returns which layers failed.

Failure stresses and ply stresses must be in the same orientation.

> **Warning:** This method does not take interaction of stresses in different directions in account.

Parameters

- **stress** (*list*) – A list containing the stress vector of the bottom, middle and top of each layer.
- **sl_max** (*float*) – Maximum tensile stress in longitudional direction.
- **sl_min** (*float*) – Maximum compressive stress in longitudional direction.
- **st_max** (*float*) – Maximum tensile stress in transverse direction.
- **st_min** (*float*) – Maximum compressive stress in transverse direction.
- **tlt_max** (*float*) – Maximum shear stress in material axis system.

Returns **pass** – True if the load was below the maximum allowables.

Return type bool

`failure.`**`tsai_hill`**(*stress*, *sl_max*, *sl_min*, *st_max*, *st_min*, *tlt_max*)

Test wether the stresses are outside the plane stress Tsai-Hill criteria.

Failure stresses and ply stresses must be in the same orientation. The method is an extension of the von Mieses stress criteria.

Parameters

- **stress** (*list*) – A list containing the stress vector of the bottom, middle and top of each layer.
- **sl_max** (*float*) – Maximum tensile stress in longitudional direction.
- **sl_min** (*float*) – Maximum compressive stress in longitudional direction.
- **st_max** (*float*) – Maximum tensile stress in transverse direction.
- **st_min** (*float*) – Maximum compressive stress in transverse direction.
- **tlt_max** (*float*) – Maximum shear stress in material axis system.

Returns **pass** – True if the load was below the maximum allowables.

Return type bool

`failure.`**`tsai_wu`**(*stress*, *sl_max*, *sl_min*, *st_max*, *st_min*, *tlt_max*)

Test wether the stresses are outside the plane stress Tsai-Wu criteria.

Failure stresses and ply stresses must be in the same orientation.

> **Warning:** This criteria is a bad approximation for compression failure.

Parameters

- **`stress`** (*`list`*) – A list containing the stress vector of the bottom, middle and top of each layer.
- **`sl_max`** (*`float`*) – Maximum tensile stress in longitudional direction.
- **`sl_min`** (*`float`*) – Maximum compressive stress in longitudional direction.
- **`st_max`** (*`float`*) – Maximum tensile stress in transverse direction.
- **`st_min`** (*`float`*) – Maximum compressive stress in transverse direction.
- **`tlt_max`** (*`float`*) – Maximum shear stress in material axis system.

Returns **pass** – True if the load was below the maximum allowables.

Return type bool

1.4.3 Indices and Tables

- genindex
- modindex
- search

1.5 MIT License

CHAPTER 2

Example

The example disscussed below is formed by snipplets of example.py.

At the start of the file the required packages are needed. A minimum requirement is *numpy* and the different scripts related to this project

```
# Import external packages.
import numpy as np

# Import local packages.
import homogenization
import abdcal
import deformation
```

Currently one is required to define the ply parameters in the ply axis system. In the future this will be replaced by a script where a basic homonigization is performed on the constituants of each ply.

```
# List the elastic properties of the ply.
El = 142 * 1e3  # MPa
Et = 13 * 1e3  # MPa
G = 5 * 1e3  # MPa
nult = 0.3  # -

# List the failure properties of the ply.
Xt = 2200  # MPa
Xc = 1850  # MPa
Yt = 55  # MPa
Yc = 200  # MPa
Smax = 120  # MPa

# List the other properties of the ply.
t = 0.16  # mm

# Calculate the ply stiffness matricess matrix.
Q = abdcal.QPlaneStress(El, Et, nult, G)
```

Then the laminate properties musth be defined. Starting with the stacking sequence which consists of a list of the rotation angles of each ply (global to ply axis system) and a list with the thickness of each ply and the ply stiffness matrix Q. All list must be orderd from the top to the bottom of the laminate. Notice that the positive z direction is downward by convention.

Afterwards the ply properties can be used to calculate the ABD matrix and its inverse.

```
# Define the stacking sequence.
angles_deg = [0, 0, 45, 90, -45, -45, 90, 45, 0, 0]
thickness = [t] * len(angles_deg)
Q = [Q] * len(angles_deg)

# Calculate the ABD matrix and its inverse.
abd = abdcal.abd(Q, angles_deg, thickness)
abd_inv = abdcal.matrix_inverse(abd)
```

Now a load or deformation vector can be applied. Here the load vector was used. The load vector is a 1 by 3 numpy matrix consists of the running loads and moments in the form of $(N_x, N_y, N_{xy}, M_x, M_y, M_{xy})^T$. Similarly a deformation vector is defined as $(\varepsilon_x, \varepsilon_y, \varepsilon_{xy}, \kappa_x, \kappa_y, \kappa_{xy})^T$. Afterwards the resulting ply stresses and loads (in their local axis system) can be calculated. The strain and stress calculations are performed at the top and bottom of each ply. Detials can be found in the documentation of the deformation module.

```
# Calculate the deformation caused by a given running load.
NM = np.matrix([0, 1, 0, 1, 0, 0]).T  # MPa/mm and MPa*mm/mm
deformed = deformation.load_applied(abd_inv, NM)

# Calculate the stress in each layer caused by the running loads.
strain = deformation.ply_strain(deformed, Q, angles_deg, thickness)
stress = deformation.ply_stress(deformed, Q, angles_deg, thickness, plotting=True)
```

Lastly the stresses are used to calculate if the failure criterias are violated. Here the the max stress, Tsai-Wu and Tsai-Hill criterias are used. It is reccomended that the user reads up on the differences between the possible criteria, all of them have their specific strength and weaknesses and are meant for their specif purpose. If one does not keep this in mind properly one will end up with flawed designs.

```
# Testing whether the failure criterias are violated.
failure.max_stress(stress, Xt, Xc, Yt, Yc, Smax)
failure.tsai_wu(stress, Xt, Xc, Yt, Yc, Smax)
failure.tsai_hill(stress, Xt, Xc, Yt, Yc, Smax)
```

CHAPTER 3

Indices and Tables

- genindex
- modindex
- search

Python Module Index

Index