laminate_analysis & Shear_Flow calculate Introduction

Eacaen
tianyunhu@gmail.com

2017 年 6 月 20 日

Contents

1 Composite Material Calculation with CLT

1.1 class Fibre

The class designed to define and change the fibre material's properties.

1.1.1 eg.

```
f = Fibre(Ef1=74000,Ef2=74000,Gf12=30800,\
        vf21=0.2,density=2.55)
```

1.2 class Matrix

The class designed to define and change the matrix material's properties.

1.2.1 eg.

```
m = Matrix(Em=300,Gm=1222,vm=0.35,density=1.18)
```

1.3 class Lamina

The class is used to define the signal lamina, its properties and engineering constants.

In the lamina initialization, you need to give the Elastic moduli E_1, E_2, the shear moduli G_{12}, the major Poisson's ration v_{21} or v_{12}. And if you want to do the strength failure analysis, you need to give the tensile strength and compressive strength parallel to the fibre X_t, X_c, the tensile strength and compressive strength of the unidirectional layer transverse to the fibre Y_t, Y_c, and the shear strength S_{21}. Don't forget to define the angle and thickness of the lamina.(All the constants' default value are zero 0)

And the second way to define from the fibre and matrix materials have been defined above, read the example to get more.

1.3.1 eg.

```
a=Lamina(E1=5.4e4,E2=0.001,G12=0.001,v21=0.25,
Xt=1.05e3,Xc=1.05e3,Yt=28,Yc=140,S=42,
angle=0,thickness=1)
```

After define the lamina,you can get the matrix of the lamina,like matrix_Q or matrix_Qbar, you can get more if you look for the source code.

1.4 class Laminate

The class is used to define the Laminate.

After the laminate initialization, you need to add the lamina as the lay up order of the laminate order.Then update the laminate you define,you can get the matrix_ABD, the thickness and so on.

1.4.1 eg.

```
LA = Laminate()   #laminate initialization
LA.add_Lamina(a) #add lamina
LA.add_Lamina(a)
LA.update()       #update and start calculation
LA.repalce_Lamina(0,b)
#replace the first(start form 0) lamina_a with the lamina_b
LA.remove_Lamina(0) #remove the first lamina
LA.update() #need to update again when change the lamina
```

1.5 class Loading

The class is used to define the Loading apply to the laminate.

After the laminate update, you can apply the load defined to the laminate, and get the stress and strain σ_1, σ_2,τ_{12} in the lamina COS, the stress and strain σ_x, σ_y,τ_{xy} in the laminate COS.

1.5.1 eg.

```
Load = Loading(1,0,0,0,0,0) #load define
Load.apple_to(LA) # load apply to laminate
Load.change_Load(2,0,0,0,0,0) #change load
Load.apple_to(LA) # load apply to laminate again
```

1.6 class Failure_Criterion

The class is used text the strength of the laminate use Failure_Criterion.You can choose which lamina to test or whole laminate.

1.6.1 eg.

```
c=Failture_Criterion()  #Criterion initialization
c.Tsai_Hill(Load,layer_num=1)
#choose the strength criterion and which lamina or all
r= c.ret_list #get the answers
```

1.7 laminate_Tools Introduction

This part can print, save and plot the results in excel file format, you should give a name of the results you want to save.(the default do not save,and the name without '.' in). When calling the function, you can also choose which lamina's results to show or show all the laminate.

1.7.1 eg.

```
print Report_strain(Load,mode='12')
plot_strain(Load,mode='xy',max_ten=None,mode2='1')
```

2 Shear Flow Calculation

2.1 graph

Basic data structure, record the points connection relationship by vertexes and edges. You can add/remove vertexes and edges, get/change weight between edges.

2.2 read_exe

the class used to read the coordinate value from the target Excel and return the value in the format [[...],[...]...[...]]; It can draw the points in the coordinate by use the function *draw_points*

2.2.1 eg.

```
val = Read_COS('path')  #read the Excel
draw_points(val.cos_value)#plot the points
```

2.3 graph_plug-in

The graph's plug-in to help draw to show connection between points,offer some functions to convert data format between list and direction, collect the weight as thickness in a open line in the graph.

2.3.1 eg.

```
draw_point_graph(graph,dir,[...])
        #draw the points connected
cos_To_dict(...) and dict_To_cos(...)
        #convert the coordinate value between list and direction
graph_line_thickness(...)
        # get the thickness distribution in a open line
```

2.4 Profile_Constant

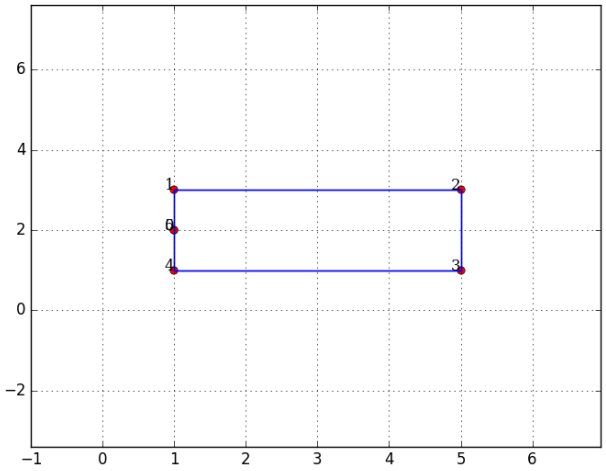
the class used to calculate the whole profile's engineer constant(open or closed).It can accept the parameter in the format coordinate value or a graph and return the *second moment area* I_x, I_y, I_{xy} and the area.

2.4.1 eg.

```
Profile_Constant(.. / ..)
#initialize the profile and calculate
Find_Centroid(...)#return the centroid cos_value
```

2.5 Open_profile

The open_profile can only calculate the open shape in a line like below:



the order of the points: 0-1-2-3-4-5(0)

accept the parameter :

- 1_{st}.coordinate value or vertices's coordinate value connected in open line in a graph,the value list like:[[...], [...]...[...]];
- 2_{nd}.the profile's engineer constant,I_x, I_y, I_{xy}.
- 3_{rd} $4_{th}$are the shear force Q_x, Q_y.

2.5.1 eg.

```
Profile_Constant(....) #prepare the constant value firstly
Open_Profile(.../...) #calculate the shear flow the line
```