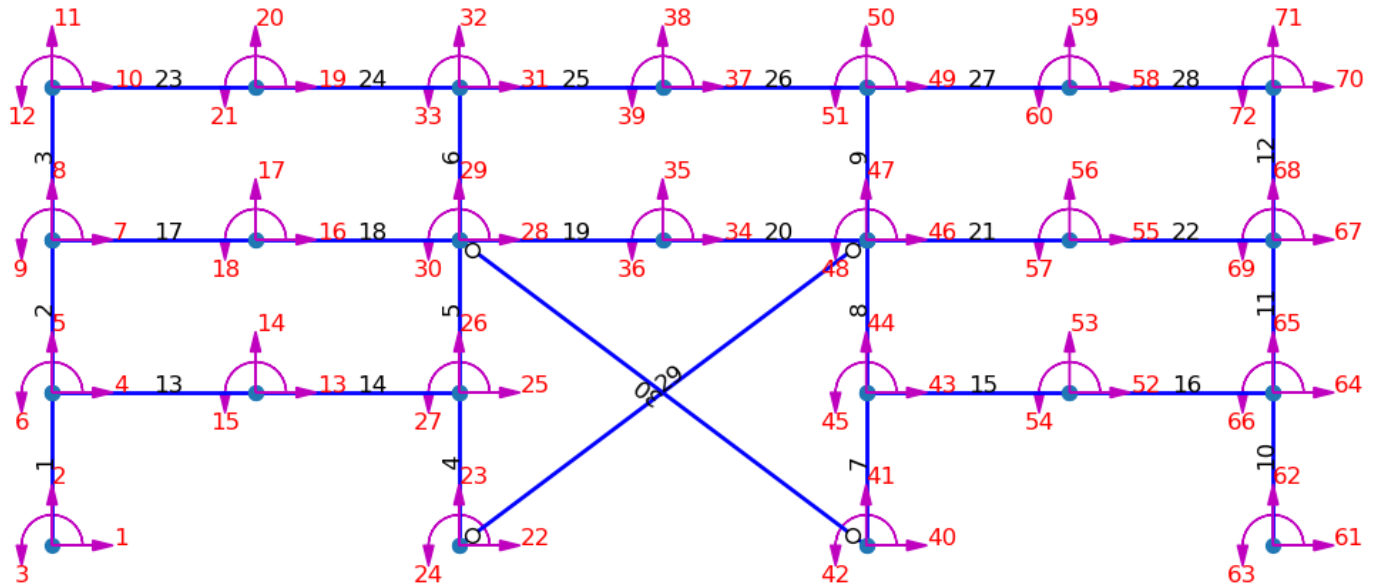


2DFrame - Python



Phạm Hoàng Anh

Bộ môn Cơ học kết cấu

Hà nội, 08 - 2024

Tính năng

Phân tích tĩnh hệ thanh phẳng bằng PTHH

- Phần tử dàn phẳng (kéo/nén)
- Phần tử thanh phẳng (kéo/nén + uốn)
- Tải tập trung tại nút, phân bố đều trên phần tử
- Chuyển vị cưỡng bức gối tựa
- Gối đàn hồi

Tiện ích

Biểu diễn đồ họa

- Sơ đồ hình học
- Sơ đồ rời rạc hóa
- Sơ đồ biến dạng
- Phản lực liên kết
- Biểu đồ nội lực

Tiện ích

Biểu diễn bảng kết quả

- Phản lực liên kết
- Chuyển vị nút
- Nội lực trong phần tử

Hoạt động

Hoạt động trên môi trường Python

- Câu lệnh trực tiếp
- Script (chương trình): *kiến nghị sử dụng phương thức này*

Cài đặt

- Python và trình soạn thảo: *kiến nghị dùng Anaconda và Spider*

<https://www.anaconda.com/>

- CALFEM – Python:

```
pip install calfem-python
```

- Module_2DFrame.py

Cấu trúc chương trình

Nạp các thư viện

Khởi tạo mô hình tính toán

Nhập các bảng số liệu

Thực hiện phân tích kết cấu

Biểu diễn kết quả

Cấu trúc chương trình

Nạp các thư viện

```
2 import matplotlib.pyplot as plt
3 import Module_2DFrame as st
```

Tên thư viện




Tên viết tắt

Cấu trúc chương trình

Khởi tạo mô hình tính toán

```
6  
7 model = st.SystemModel('2Dframe')
```



Loại kết cấu:
'2Dframe'
'2Dtruss'

Cấu trúc chương trình

Nhập các bảng số liệu

Số liệu về nút:

```
model['Node'] = [  
    [x1, y1],  
    [x2, y2],  
    ...  
    [xn, yn],  
]
```

Tọa độ nút 1

Tọa độ nút 2

Tọa độ nút n

Cấu trúc chương trình

Nhập các bảng số liệu

Số liệu về phân tử:

```
model['Ele'] = [  
    [i, j, m, si, sj],  
    ...,  
]
```

Nút đầu

Nút cuối

Tiết diện

Mã liên kết
nút cuối

Mã liên kết nút đầu:
0 – khớp; 1 – Hàn

Cấu trúc chương trình

Nhập các bảng số liệu

Số liệu về tiết diện:

```
model['Mat'] = [  
    [E1, A1, I1],  
    [E2, A2, I2],  
    ...,  
    [Em, Am, Im]  
]
```

Tiết diện 1

Tiết diện 2

Tiết diện m

Mô đun
đàn hồi

Diện tích

Mô men
quán tính

Cấu trúc chương trình

Nhập các bảng số liệu

Số liệu về liên kết:

```
model['Bound'] = [  
    [node, c1, c2, c3],  
    ...,  
]
```

Tên nút

Liên kết phương ngang:
1 – ngăn cản; 0 – tự do

Liên kết xoay

Liên kết
phương đứng

Cấu trúc chương trình

Nhập các bảng số liệu

Số liệu tải trọng nút:

```
model['Nload'] = [  
    [node, Fx, Fy, Mz],  
    ...,  
]
```

Mô men tập
trung

Tên nút

Lực tập trung
phương ngang

Lực tập trung
phương đứng

Cấu trúc chương trình

Nhập các bảng số liệu

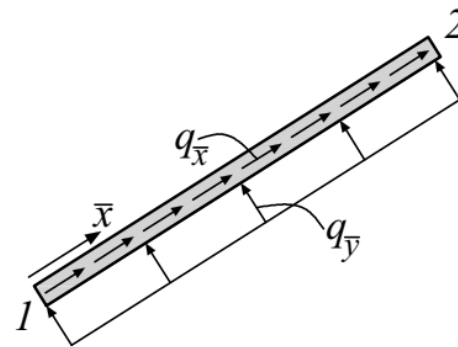
Số liệu tải trọng phân tử:

```
model['Eload'] = [  
    [ele, qx, qy],  
    ...,  
]
```

Tên
phần tử

Lực phân bố dọc
trục thanh

Lực phân bố vuông
góc trục thanh



Cấu trúc chương trình

Nhập các bảng số liệu

Số liệu chuyển vị liên kết:

```
model['Supp'] = [  
    [node, Zx, Zy, Zz],  
    ...,  
]
```

Tên nút

Chuyển vị
phương ngang

Chuyển vị
phương đứng

Chuyển vị
xoay

Cấu trúc chương trình

Nhập các bảng số liệu

Số liệu gói đàn hồi:

```
model['Spring'] = [  
    [node, kx, ky, kz],  
    ...,  
]
```

Độ cứng xoay

Tên nút

Độ cứng gói
phương ngang

Độ cứng gói
phương đứng

Cấu trúc chương trình

Thực hiện phân tích kết cấu

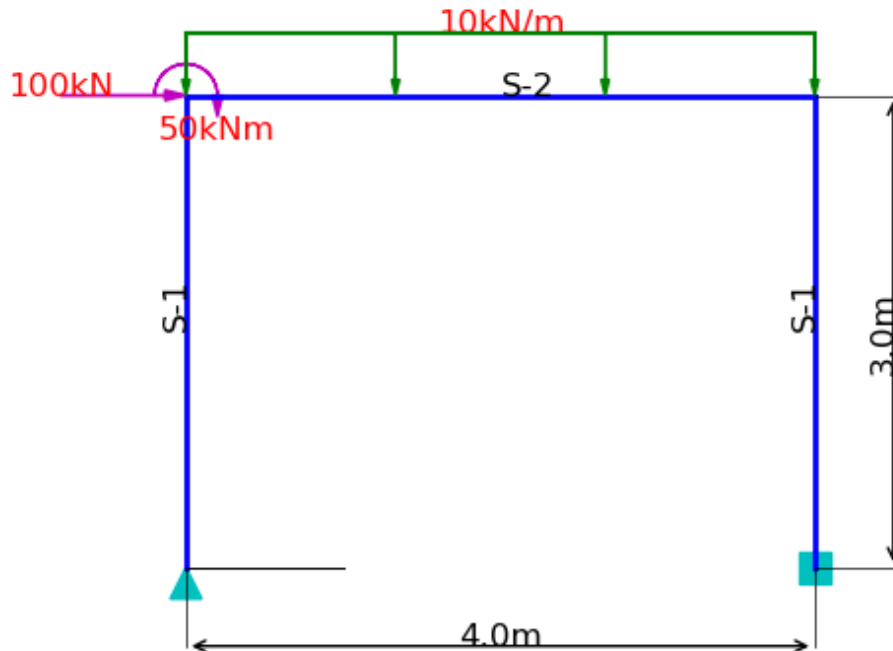
Khung phẳng:

```
model = st.Solve2Dframe(model)
```

Dàn phẳng:

```
model = st.Solve2Dtruss(model)
```

Biểu diễn đồ họa

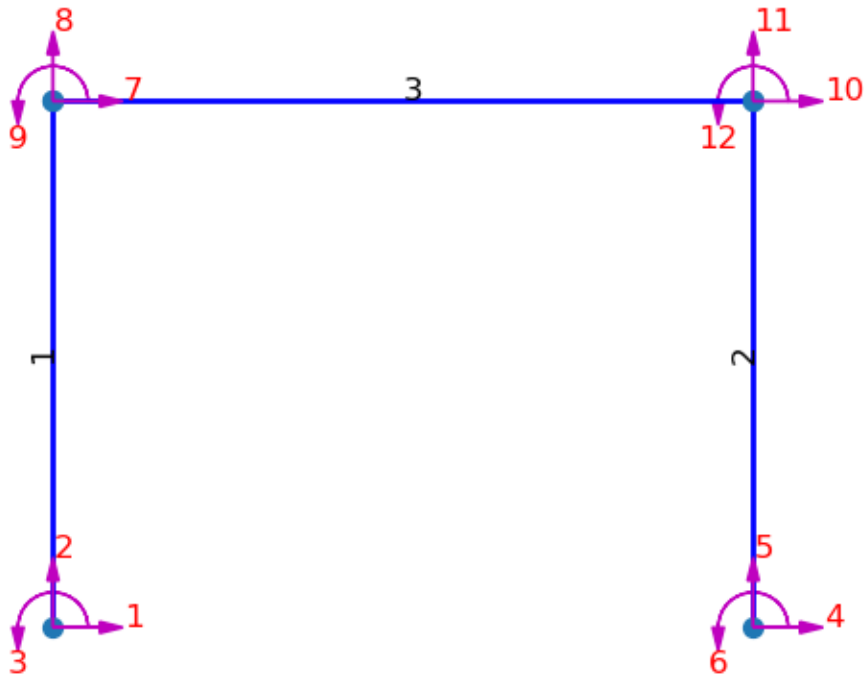


Lệnh vẽ Sơ đồ hình học:

```
show_geometry(model,  
               etype='E',  
               ID=None,  
               show_node=False,  
               show_ele=False)
```

```
st.show_geometry(model, etype='S-', show_ele=True)  
st.draw_dim([0,0,0], [L,0,0], offset=-0.5)  
st.draw_dim([L,0,0], [L,H,0], offset=-0.5)
```

Biểu diễn đồ họa

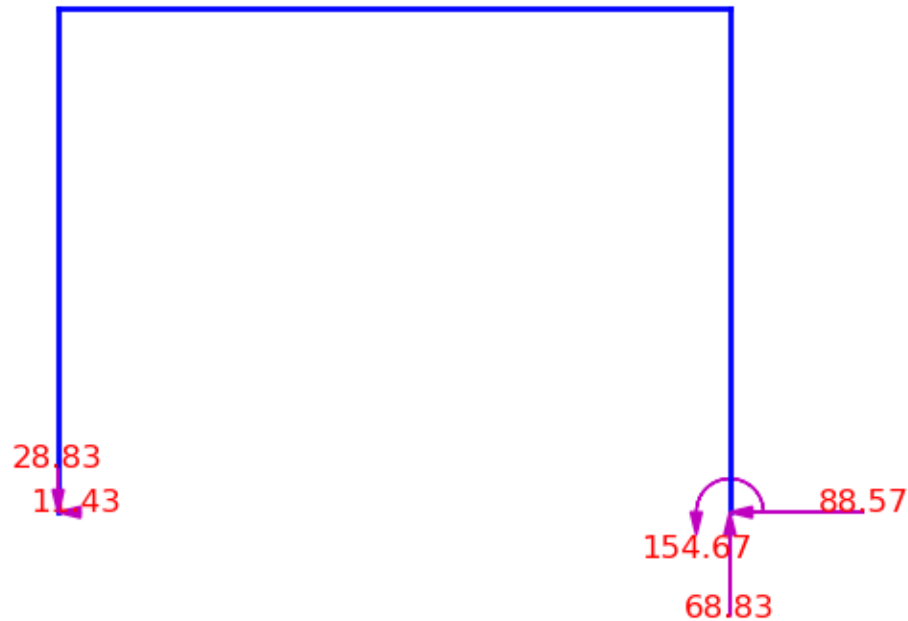


Lệnh vẽ Sơ đồ rời rạc:

```
show_FEM(model,  
          scale='Auto',  
          show_node=False)
```

```
st.show_FEM(model)
```

Biểu diễn đồ họa

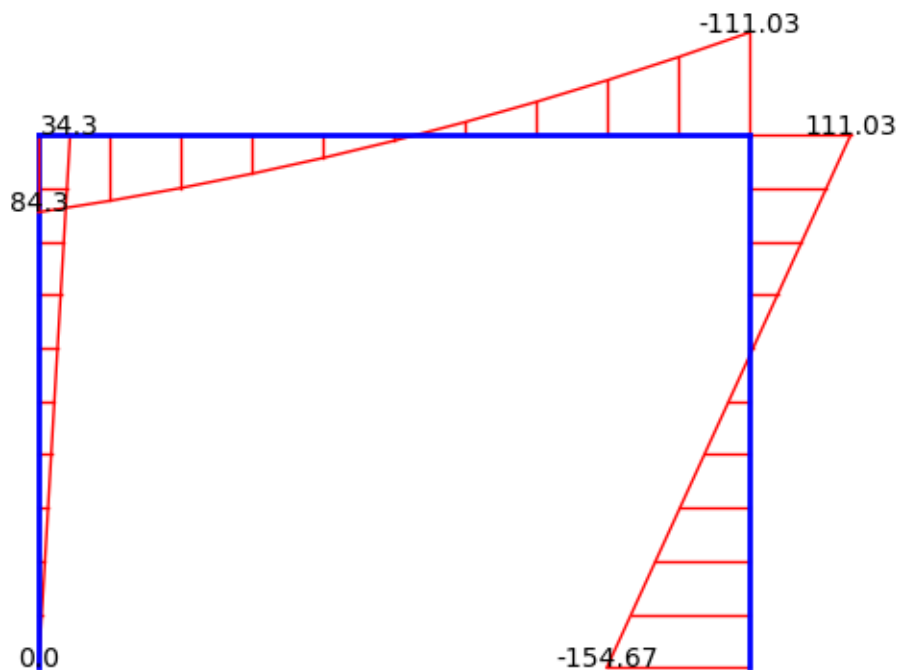


Lệnh vẽ Phản lực liên kết:

```
show_reaction(model,  
scale='Auto')
```

```
st.show_reaction(model)
```

Biểu diễn đồ họa



Lệnh vẽ Biểu đồ nội lực:

```
show_moment(model,  
             scale='Auto')
```

```
show_shear(model,  
            scale='Auto')
```

```
show_axial(model,  
            scale='Auto')
```

```
st.show_moment(model)
```

Biểu diễn đồ họa



Lệnh vẽ Sơ đồ biến dạng:

```
show_displacement(model,  
scale='Auto')
```

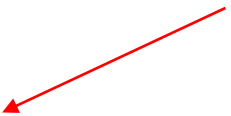
```
st.show_displacement(model)
```

Biểu diễn kết quả dạng bảng

Phản lực liên kết:

`disp_react(model)`

Danh sách
nút



Chuyển vị nút:

`disp_ndisp(model, nlist=None)`


Nội lực trong phần tử:

`disp_eforce(model, elist=None)`

Ma trận độ cứng và véc tơ lực nút của phần tử:

`disp_ematrix(model, elist=None, coord='Global')`

Danh sách
phần tử



'Global'
'Local'



Trích xuất dữ liệu

Chuyển vị nút:

```
model [ 'disp' ]
```

```
array([[ 0.00000000e+00,  0.00000000e+00, -5.56096613e-05],  
       [ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00],  
       [ 1.42331477e-04,  1.64751030e-07, -3.11121549e-05],  
       [ 1.41656675e-04, -3.93322459e-07, -3.11750473e-05]])
```

dx

dy

fi

```
model [ 'disp' ] [2]
```

Nút 3

```
array([ 1.42331477e-04,  1.64751030e-07, -3.11121549e-05])
```

```
model [ 'disp' ] [2,0]
```

```
0.00014233147745030067
```

Trích xuất dữ liệu

Nội lục:

Phần tử 2

`model['force'][1]`

```
array([[ -68.83143033,  88.56783039, -154.67427869],
       [ -68.83143033,  88.56783039, -128.10392957],
       [ -68.83143033,  88.56783039, -101.53358046],
       [ -68.83143033,  88.56783039,  -74.96323134],
       [ -68.83143033,  88.56783039,  -48.39288222],
       [ -68.83143033,  88.56783039,  -21.82253311],
       [ -68.83143033,  88.56783039,   4.74781601],
       [ -68.83143033,  88.56783039,  31.31816512],
       [ -68.83143033,  88.56783039,  57.88851424],
       [ -68.83143033,  88.56783039,  84.45886336],
       [ -68.83143033,  88.56783039, 111.02921247]])
```

N

Q

M

Trích xuất dữ liệu

Phản lực liên kết:

`model['res']`

```
array([[ -1.14321696e+01,  -2.88314303e+01,   7.10542736e-15],  
       [ -8.85678304e+01,   6.88314303e+01,   1.54674279e+02]])
```

R_x

R_y

M

Áp dụng

2DFrame hỗ trợ hiệu quả trong giảng dạy các môn học:

- Cơ học kết cấu 1
 - Cơ học kết cấu 2
 - Phần tử hữu hạn
-
- Làm đề bài, đáp án: linh hoạt, dễ tùy biến
 - Vẽ hình minh họa: sơ đồ kết cấu, biểu đồ nội lực, ...

<https://github.com/HoangAnh-Pham/2DFrame>