

Ejemplo2

May 11, 2024

```
[1]: from openseespy.opensees import *
import opsv as opsv

import numpy as np
import matplotlib.pyplot as plt
```

```
[10]: #-----
# Modelado y análisis estático de una armadura.
# Unidades del modelo: [N,m]
#-----
#Eliminar cualquier modelo existente
wipe()

#ndm=2 dimensiones y ndf=2 grados de libertad por nodo
model('basic', '-ndm', 2, '-ndf', 2)

#-----
#2 OpenSeesPy - Definición del modelo estructural
#-----

# Se establecen los nodos del modelo y sus respectivas restricciones.
# Solo se brindan dos coordenadas ya que el modelo es en 2D.(ID nodo, X, Y)
node(1, 0.0, 0.0)
node(2, 2.3, 3.0)
node(3, 2.3, 0.0)

# Definición de condiciones de contorno, (Fx, Fy, M)
# 1 - Grado de libertad restringido.
# 0 - Grado de libertad no restringido.
fix(1, 0, 1, 0)
fix(2, 1, 1, 0)

# Se define el material de los elementos (Tipo, ID mat, E)
uniaxialMaterial("Elastic", 1, 3000.0)

# Tipo de elemento (Tipo,ID elem, i, j, A, E, Iz, ID mat)
#Element: Truss, elasticBeamColumn,
element("Truss",1,1,2,10.0,1)
```

```

element("Truss",2,1,3,5.0,1)
element("Truss",3,2,3,5.0,1)
#-----
#2 OpenSeesPy - Definición de carga
#-----
# 3 pasos, (serie temporal->patrón de carga->cargas)

# Serie de tiempo (Tipo, ID st)
#Tipo: Constant, lineal,
timeSeries("Constant", 1)

# patrón de carga (Tipo, ID pc, ID st)
#Tipo: UniformExcitation, MultipleSupport,
pattern("Plain", 1, 1)

# Cargas (nodeID, xForce, yForce)
load(3, 22.0, 0.0)

# -----
#3 OpenSeesPy - Análisis
# -----
system("BandSPD")

# create DOF number
numberer("RCM")

# create constraint handler
constraints("Plain")

# create integrator
integrator("LoadControl", 1.0)

# create algorithm
algorithm("Linear")

# create analysis object
analysis("Static")

# perform the analysis
analyze(1)

# -----
#4 OpenSeesPy - Resultados
# -----
n=3
for i in range(n) :
    ux = round(nodeDisp(i+1, 1),5) #Horizontal nodal displacement

```

```

    uy = round(nodeDisp(i+1, 2),5) #Vertical nodal displacement
    print(f'Node {i+1}: Ux = {ux} m, Uy = {uy}')

mbrForces = np.array([])
for i, mbr in enumerate(members):
    axialForce = round(basicForce(i+1)[0]/1000,2)
    mbrForces = np.append(mbrForces,axialForce) #Store axial loads
    print(f'Force in member {i+1} (nodes {mbr[0]} to {mbr[1]}) is {axialForce} kN')

# -----
# OpenSeesPy - Gráficos
# -----

opsv.plot_model()

opsv.plot_loads_2d()

opsv.plot_defo()

sfacN, sfacV, sfacM = 8.e-3, 5.e-5, 5.e-5

opsv.section_force_diagram_2d('N', sfacN)
plt.title('Axial force distribution')

#opsv.section_force_diagram_2d('T', sfacV)
#plt.title('Shear force distribution')

#opsv.section_force_diagram_2d('M', sfacM)
#plt.title('Bending moment distribution')

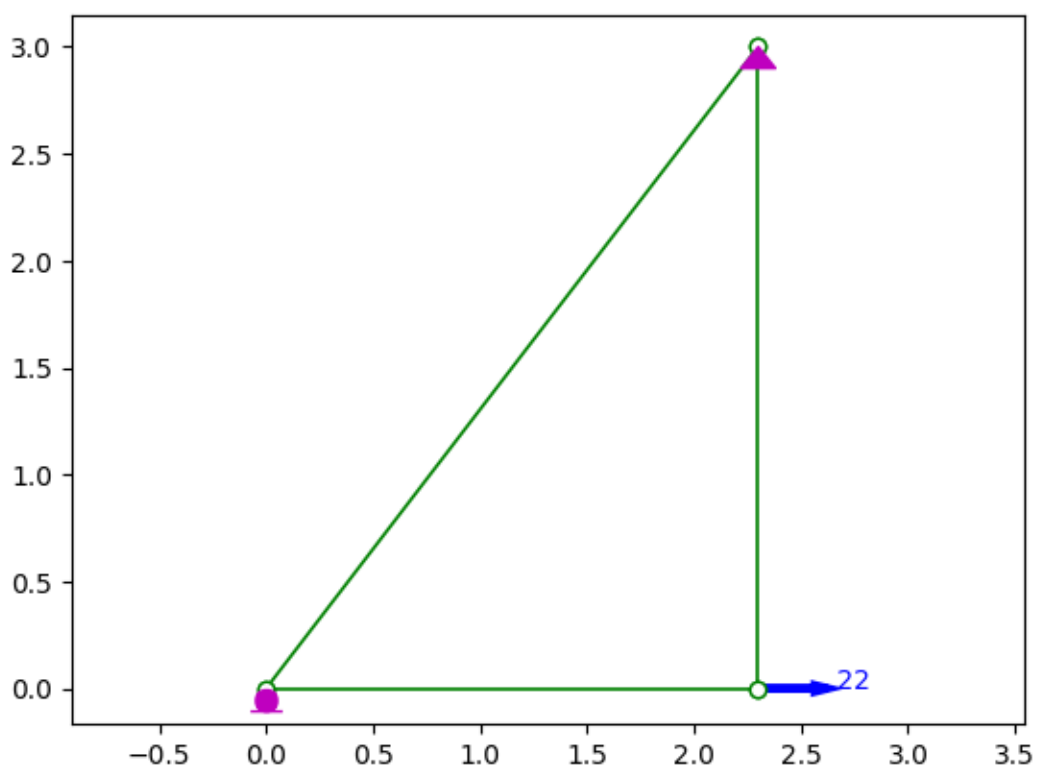
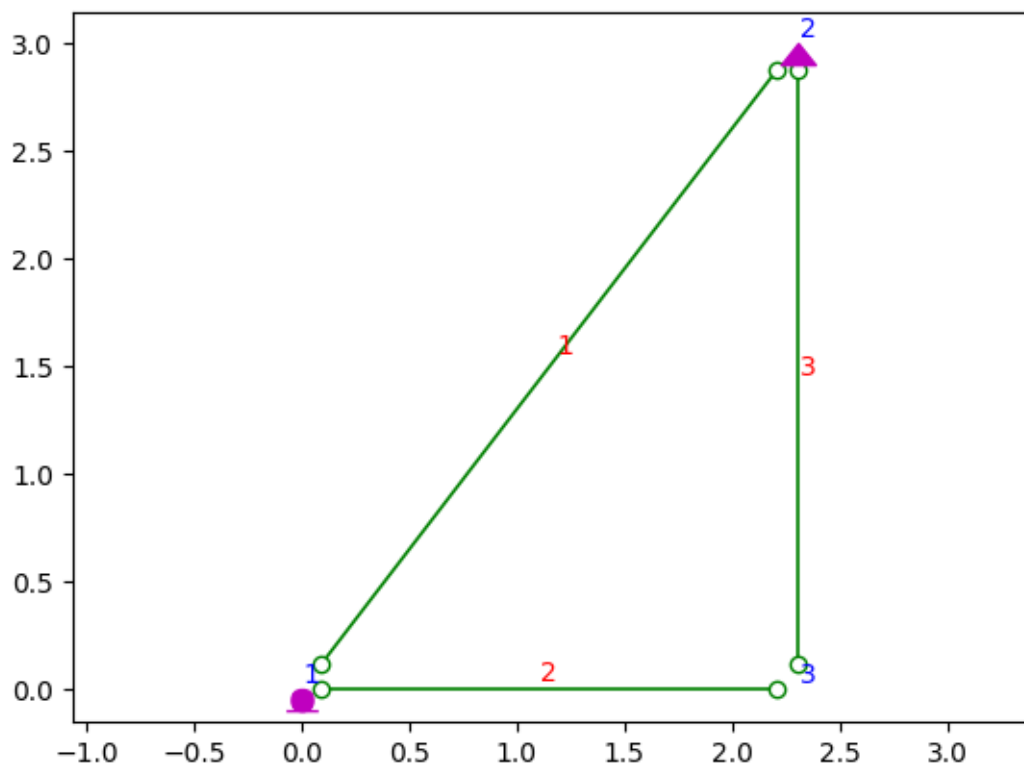
plt.show()

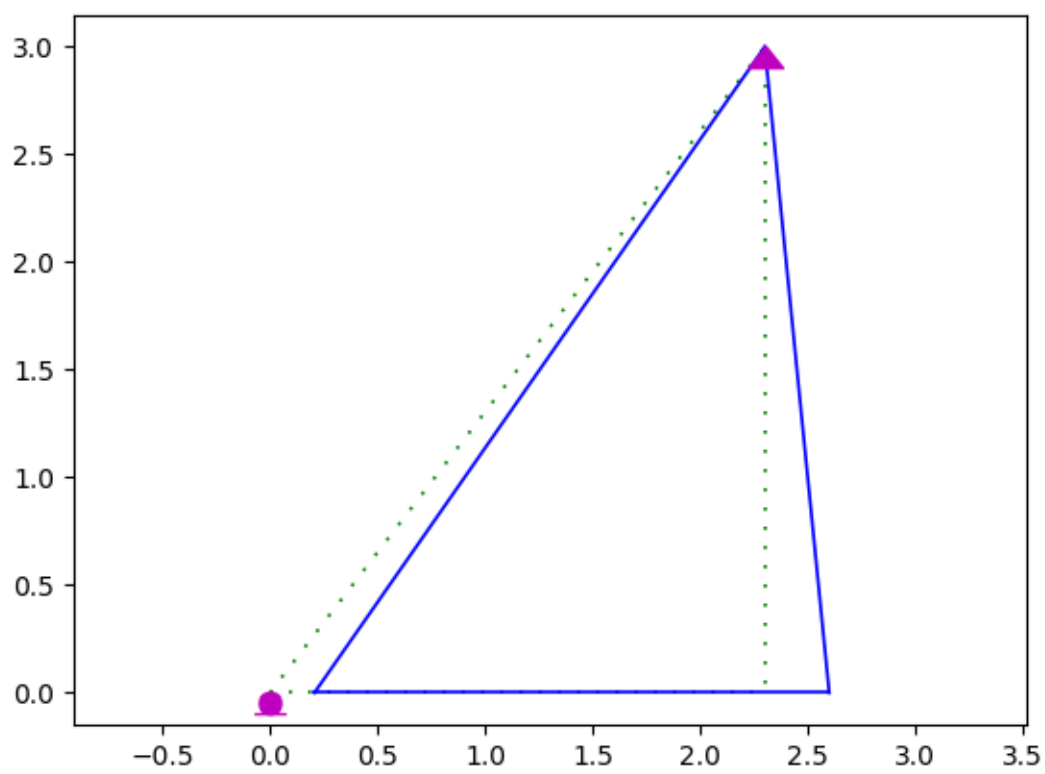
```

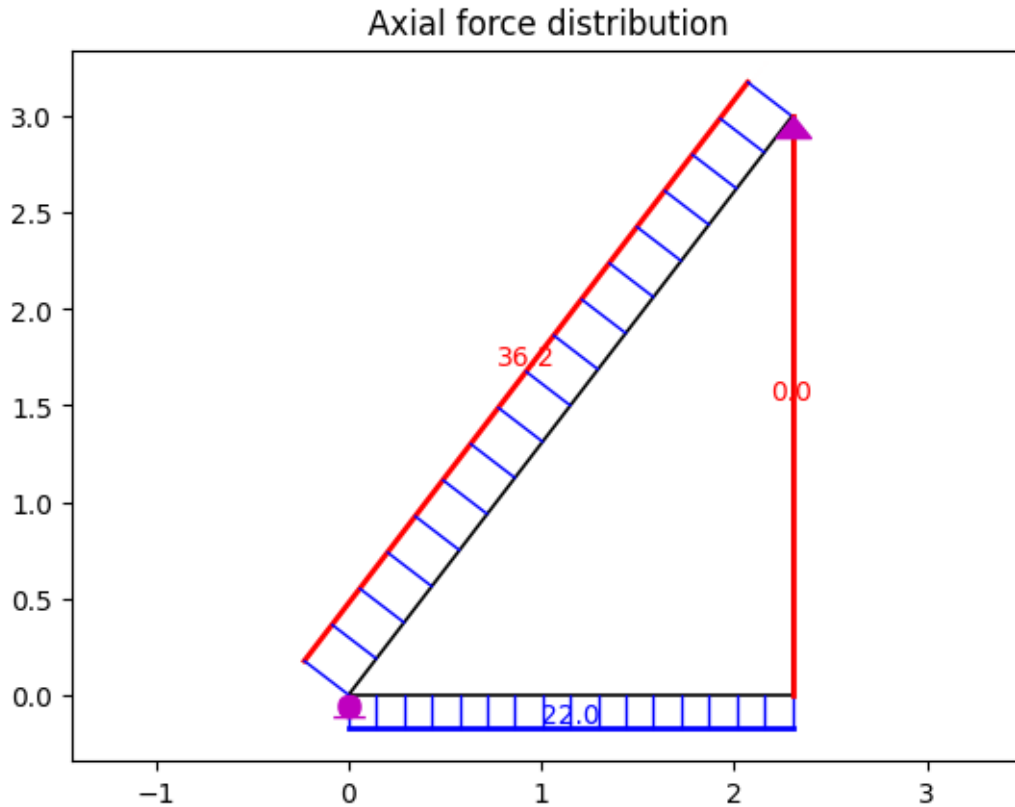
Node 1: Ux = 0.00749 m, Uy = 0.0

Node 2: Ux = 0.0 m, Uy = 0.0

Node 3: Ux = 0.01086 m, Uy = 0.0







1 1. Configuración básica

This is bold text

Comenzamos simplemente definiendo el módulo de Young del material y el área de la sección transversal de los miembros de celosía como dos constantes.

```
[9]: E = 200*10**9 #(N/m^2)
     A = 0.005 #(m^2)
```

1.1 1.1 Configuración básica

El comando model

```
[7]: #Eliminar cualquier modelo existente
     wipe()

     #ndm=2 dimensiones y ndf=2 grados de libertad por nodo
     model('basic', '-ndm', 2, '-ndf', 2)
```

Definimos los miembros o conectividad