

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Mar  9 11:24:56 2022

@author: juanmeriles
"""
```

```
import sys as sys
import random as random
import math as math
import time as time

def main():
    # 0 0
    # 0 4
    # 0 8
    # 4 0
    # 4 4
    # 4 8
    # 8 0
    # 8 4
    # 8 8
    # 12 0
    # 12 4
    # 12 8

    # 0 0
    # 0 4
    # 0 8
    # 4 0
    # 5 3
    # 4 8
    # 8 0
    # 9 5
    # 8 8
    # 12 0
    # 12 4
    # 12 8

    # 1 4 5 2
    # 2 5 6 3
    # 4 7 8 5
    # 5 8 9 6
    # 7 10 11 8
    # 8 11 12 9

    # 1 1 1 1 0 1 1 0 1 1 1 1
```

```
class element
```

```
    def __init__(self)
        pass
```

```
def ShapeFcn
```

```
    1 4 1 1
    1 4 1 1
    1 4 1 1
    1 4 1 1
        0 0 0 0
        0 0 0 0
```

```
    return
```

```
def Jacobian
```

```
    1 4 1 0 0 1 1 0 1 2 0 1
    1 4 1 0 1 1 1 1 1 2 1 1
    1 4 1 0 0 1 1 0 1 2 0 1
    1 4 1 0 1 1 1 1 1 2 1 1
```

```
    return
```

```
def plotElements
```

```
    for i in range(len
```

```
        for j in range(len 0
```

```
        1 0
        1 1
```

```
        0
        0
```

```
def func
```

```
    1 4 1 0 1 1
    1 4 1 0 1 1
    1 4 1 0 1 1
    1 4 1 0 1 1
```

```

        0 0 0 0 0
    return
def whichel
    if 4 and 4
        1
    elif 4 and 8
        2
    elif 8 and 4
        3
    elif 8 and 8
        4
    elif 12 and 4
        5
    elif 12 and 8
        6
    else
        1
        print 'error'
    return

```

```

#def GaussInt():

```

```

    len
for in range

```

```

        0 1          1 1
    0 1          1 1          2
        0 0          0 1
        2 0          2 1

```

```

for in range

```

```

        0 1          1 1
    0 1          1 1

```

```

1 1 1 1 1 1 1 1 1 1 1 1
0 8 16 4 12 20 8 16 24 12 20 28
2 8 7 6 1 1 3 2 4 4 3 0

```

```

12 12
12 1
12 1
12 1
0
for i in range
3 5 0      3 5
5 9 8 9 5 9
for i in range len
for j in range len
2 0 0      2 0 1
0 0      0 2
2 0      0 2
0 1
0 1
0
1
2
#which element are we in
#print(inel)
#find natural coords of gp
0 0
#print(gp_global)
#get interpolated values
-1 0 1      -1 1 1
#print(inel)
#print(natcoords)
#print(ztemp)
0 1
+1

```

```

        for i in range(len
            1
            1
            1
            for i in range(len
                1
                0
                1
                1
                1
            )

#Solve least squares

#Grab the z on corresponding natural coords
        5 3
        9 5
        1 1
        0 0
        0 0
    for i in range(len
        0
        0 0
        0
        -1 0 1
        0 2
        0
        1
        -1 1 1
        1
        -1 2
        1
        1 1 1 1 1
        0 1 1
        0 20 8
        0 0 0 1 3
        1 1 1 1 1
        1 24 12 20 28
        1 0 0 4 4 3 0
        0
        0
        0
        0
        0
        0
    for i in range
        3 5 0
        3 5
        5 9 8 9 5 9
        for i in range(len
            for i in range(len
                0 0
                2 0
                0 0
                0 1
                2 1
                0
            )

```

```

2 0 2
0
1
0
1
1
0
2
1
#which element are we in
#print(inel)
#find natural coords of gp
0 0
-1
#print(gp_global)
#get interpolated values
-1 0 1 -1 1 1
#print(inel)
#print(natcoords)
#print(ztemp)
0 1
#projected z at gauss point
0 1 1 1
0 1 1 1
0 1 1 1
0 0 0
#print(zgpap)
0 0 0
0 0 0
#Collocated z at gauss point
0 1 1 1
0 1 1 1
0 1 1 1
0 0
0 0
0 0
2
2

```

2
2
2
2

```
# zmesh1og = np.hstack(zmesh1og)
# zmesh2og = np.hstack(zmesh2og)
# zmesh3og = np.hstack(zmesh3og)
# zmesh1trans = np.hstack(zmesh1trans).flatten()
# zmesh2trans = np.hstack(zmesh2trans).flatten()
# zmesh3trans = np.hstack(zmesh3trans).flatten()
# x1 = np.hstack(x1)
# y1 = np.hstack(y1)
# ax = plt.axes(projection = '3d')
# #x1 = np.array(x1)
# #x2 = np.array(x2)
# ax.plot_trisurf(x1,y1,zmesh2og)
# ax.plot_trisurf(x2,y2,zmesh2trans)
```

```
#plt.plot(x1,y1,'o')
#plt.plot(x2,y2,'o')
```

'b'

```
#plt.plot(globalGaussPointsx_og,globalGaussPointsy_og,'o')
```

```
#plt.figure(2)
#plotElements(nodes,con,'b')
#plt.plot(globalGaussPointsx_trans,globalGaussPointsy_trans,'o')
```