Structural Analysis HW 3

Cameron Cummins (csc3323)

February 2023

1 Introduction

I chose to write my 2D finite element program in Python and can be utilized by executing the **finite_element_2d.py** in a package environment with at least **Python v3.8.8** and **Numpy v1.20.1** installed. The input file is in the JSON file format to avoid text-parsing bugs and organize the input data following an enforceable standard. In the future, if simple text files are supplied as input, a pre-processing script could be developed to convert those text files into JSON files to avoid modifying the core parsing methods of the finite element code.

The code itself is organized into a group of functions contained within the **FiniteElement2D class**. Each function computes a stage of the finite element process for 2D trusses. You can find the function calls and variables created in the *constructor*:

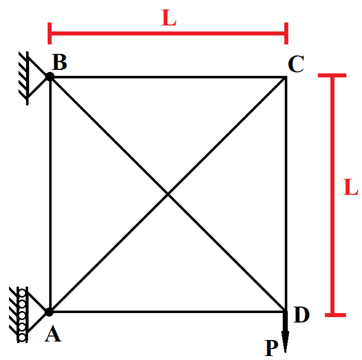
```
def __init__(self, input_file_path: str) -> None:
```

Note that Python doesn't have type declarations, but it does have type *hinting*. Here, the class constructor "hints" that the *input_file_path* parameter is of the string type and that this function does not return anything. I utilize type hints throughout the code to make it clear what functions are inputting and outputting. All functions feature a *docustring* which gives a brief overview of the functions parameters and return. Here is an example of such a docustring:

```
def buildGlobalStiffnessMatrix(self, pos: dict, ... ) -> np.array:
    """
    Constructs the global stiffness matrix using local element
        stiffness matrices.
    pos --> Nodal positions dictionary from input JSON
    conns --> Nodal connections dictionary from input JSON
    global_index --> Global index of nodal displacements to column
        position in the global stiffness matrix

    Returns a square, symmetric global stiffness matrix,
        dimensionally consistent with the global index
    """
```

2 Analyzing the Truss Problem



This truss problem translates into the following JSON in **four_node_truss.json**:

```
"node_positions": {
    "A" : [0, 0],
    "B" : [0, 1],
    "C" : [1, 1],
    "D" : [1, 0]
},
"node_connections": [
    { "labels" : ["A", "B"], "material": 1 },
    { "labels" : ["A", "C"], "material": 1 },
    { "labels" : ["A", "D"], "material": 1 },
    { "labels" : ["B", "C"], "material": 1 },
    { "labels" : ["B", "D"], "material": 1 },
    { "labels" : ["C", "D"], "material": 1 }
],
"node_xy_fixed": {
    "A" : {"u": 0},
    "B" : {"u": 0, "v": 0}
},
"external_node_forces": {
    "D": [0, -1]
}
```

Which is read into the finite element program using the following code:

```
analysis = FiniteElement2D("four_node_truss.json")
```

Here are the resulting displacements calculated by my finite element program:

$$\begin{bmatrix} A_u \\ A_v \\ B_u \\ B_v \\ C_u \\ C_v \\ D_u \\ D_v \end{bmatrix} = \begin{bmatrix} 0 \\ -0.396 \\ 0 \\ 0 \\ 0.396 \\ -1.914 \\ -0.604 \\ -2.311 \end{bmatrix}$$