

Circuit Analysis Code Documentation

Overview

This Python code is designed for analyzing electrical circuits specified in a SPICE-like format. It processes circuit elements, performs nodal analysis, and computes node voltages and current through voltage sources.

Functions

`addVsource(vElement, vlst, nodes)`

This function adds a voltage source element to the circuit.

- **Parameters:**
 - `vElement` : List containing details of the voltage source.
 - `vlst` : List of voltage source elements.
 - `nodes` : List of nodes in the circuit.
- **Returns:**
 - Updated `vlst` and `nodes` .

`addIsource(iElement, ilst, nodes)`

This function adds a current source element to the circuit.

- **Parameters:**
 - `iElement` : List containing details of the current source.
 - `ilst` : List of current source elements.
 - `nodes` : List of nodes in the circuit.
- **Returns:**
 - Updated `ilst` and `nodes` .

`addResistance(rElement, rlst, nodes)`

This function adds a resistor element to the circuit.

- **Parameters:**
 - `rElement` : List containing details of the resistor.
 - `rlst` : List of resistor elements.
 - `nodes` : List of nodes in the circuit.
- **Returns:**
 - Updated `rlst` and `nodes` .

`extractFile(filename)`

This function extracts data from a SPICE-like file.

- **Parameters:**
 - `filename` : Name of the input file.
- **Returns:**
 - Extracted data as a list.

`processFile(extractedlst)`

This function processes the extracted data to identify circuit elements.

- **Parameters:**
 - `extractedlst` : Extracted data from the file.
- **Returns:**
 - Lists of voltage sources, current sources, resistors, and nodes.

`createMatrix(vlst, ilst, rlst, nodes)`

This function creates matrices for nodal analysis.

- **Parameters:**
 - `v1st` : List of voltage source elements.
 - `i1st` : List of current source elements.
 - `r1st` : List of resistor elements.
 - `nodes` : List of nodes in the circuit.
- **Returns:**
 - Matrices `A` and `B` for nodal analysis.

`evalSpice(filename)`

This function evaluates the circuit specified in the input file.

- **Parameters:**
 - `filename` : Name of the input file.
- **Returns:**
 - Dictionary of node voltages and current through voltage sources.

Error Handling

The code handles the following errors:

- When comments are present between `'circuit'` and `'end'`
- When the resistance is zero
- Raises `ValueError` exception when resistance is negative
- Raises `ValueError` exception when the circuit is not solvable
- Raises `FileNotFoundError` exception when the input file is not valid
- Raises `ValueError` exception when the circuit format is malformed
- Raises `ValueError` exception when components other than V, I and R are found

Discussions

- Discussed about handling zero resistance with Rethesh(EE22B059)

Example usage

```
voltage_dict, current_dict = evalSpice(<path of 'circuit.ckt'>)
print(voltage_dict)
print(current_dict)
```