



# Circuitos Elétricos II

## Trabalho I - 2023.2

Universidade Federal do Rio de Janeiro  
Departamento de Engenharia Eletrônica e de Computação  
Professora: Fernanda Oliveira

Enunciado do primeiro trabalho de simulação de circuitos elétricos. Para o trabalho, os alunos devem desenvolver um programa em Python (versão 3 ou maior). Os pacotes Numpy e Scipy são permitidos. Para utilizar outros pacotes, converse com a professora. O trabalho é individual. Por favor não enviem ou recebam códigos de outros alunos.

Instruções:

1. O trabalho consiste no desenvolvimento de um programa em Python que recebe uma string com o nome de um arquivo de texto que contém a descrição de circuito por uma “netlist”, e que calcula e retorna um array do numpy com as tensões nodais do circuito calculadas através da análise nodal simples.
2. Para esse trabalho, considere que o circuito contém somente fontes de corrente independentes, fontes de corrente controladas por tensão e resistores.
3. O trabalho pode ser escrito em um ou mais arquivos “.py” (veja, abaixo, a padronização dos nomes dos arquivos) e pode usar funções ou classes, de acordo com a preferência do aluno, mas deve, necessariamente, conter uma função principal cujo nome é main. Essa função recebe por argumento uma string com o nome de um arquivo de texto que contém a “netlist” (isto é, a função deve ser definida da seguinte forma: `def main(arqNetlist)`, de forma que, ao fazer a chamada da função dentro de um “.py” o nome do arquivo deve ser colocado como uma string, ex.: `main('netlist1.txt')` ; reforçando, a função será chamada por um arquivo “.py” e não diretamente pelo terminal). A função main deve calcular e retornar um array do numpy com as tensões nodais do circuito calculadas através da análise nodal simples.
4. O array retornado pela função main deve conter uma quantidade de elementos igual ao número de nós exceto pelo terra. O elemento da posição 0 do array deve ser igual à tensão calculada para o nó 1, o elemento da posição 1 deve ser igual à tensão calculada para o nó 2, o elemento da posição 2 deve ser igual à tensão calculada para o nó 3... A numeração dos nós deve respeitar a numeração da netlist (considere que a numeração da netlist começa de zero, que é o nó terra, e que nenhum número é pulado).
5. O arquivo “.py” que contém a função main deve ter o nome seguindo o seguinte padrão: `trab1nombresobrenome.py` , onde nome deve ser substituído pelo seu primeiro nome e sobrenome pelo seu último sobrenome. Utilize somente minúsculas e não utilize caracteres especiais. Por exemplo: `trab1fernandaoliveira.py`
6. Para garantir que não haverá arquivos de alunos diferentes com o mesmo nome, os nomes de todos demais arquivos “.py” utilizados (caso o aluno utilize mais de um arquivo “.py” além daquele que contém a função main) devem conter as iniciais do aluno.

7. Exemplo: para a entrada “netlist0.txt” o valor de retorno deve ser `array([10,8])` ou `array([[10],[8]])`, onde `array` é um array do numpy.

Cada linha da netlist indica um novo componente e as informações necessárias sobre ele. A netlist pode conter comentários, que devem ser ignorados pelo programa, conforme indicado abaixo. Para resistores, fontes de corrente controladas por tensão, e fontes de corrente independentes, o seguinte padrão deve ser seguido:

- Comentários, linhas que não são usadas pelo programa: `*<comentário>`
- Resistor: `R<identificação> <nó1> <nó2> <valor da resistência>`
- Fonte de corrente controlada por tensão: `G<identificação> <nóI(fonte drena a corrente desse nó)> <nóI(fonte injeta a corrente nesse nó)> <nóv(positivo)> <nóv(negativo)> <valor da transcondutância Gm>`
- Fonte de corrente DC: `I<identificação> <nó cuja corrente é drenada pela fonte> <nó cuja corrente é injetada pela fonte> DC <valor>`

Tudo o que está entre os caracteres `<` e `>` deve ser substituído na netlist e os caracteres `<` e `>` em si também não aparecem na netlist.

Observação: a netlist costuma incluir mais informações sobre o tipo de excitação das fontes de corrente. No momento, no entanto, consideraremos somente fontes constantes. Isso será modificado mais à frente. Veja, na Figura 1, um exemplo de circuito e de netlist.

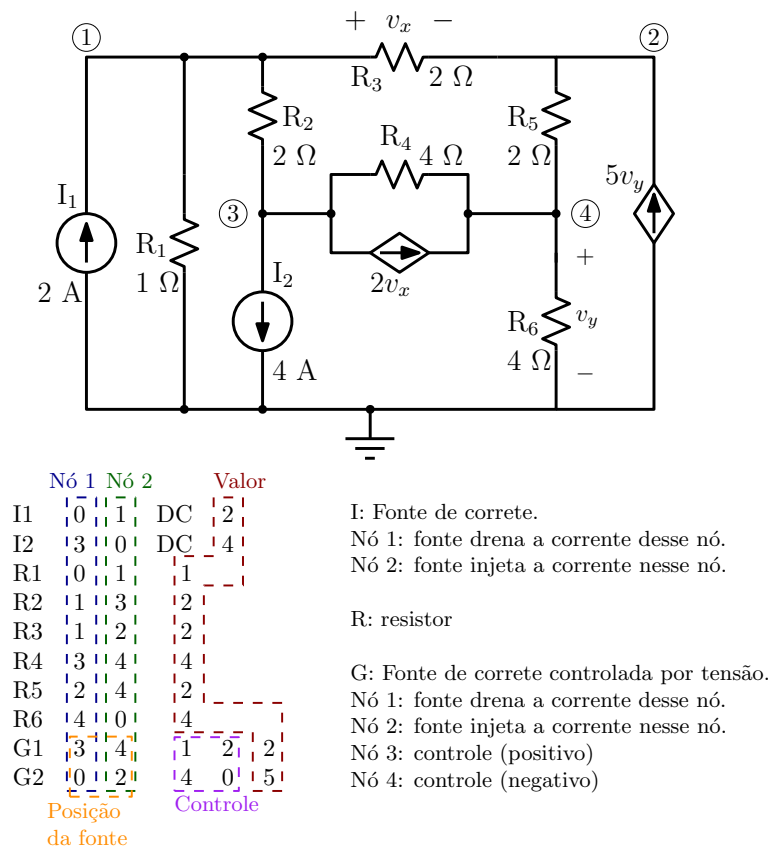


Figura 1: Exemplo de netlist.