

Enunciado do quarto trabalho de análise 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 seis argumentos, nessa ordem: uma string com o nome de um arquivo de texto que contém a descrição de circuito por uma “netlist”, um float que representa o tempo de simulação em segundos, um float que representa o passo, um float que representa a tolerância do Newton-Raphson (epsilon), uma lista de valores iniciais para cada nó da netlist (onde o valor da posição 0 é a condição inicial do nó 1), e uma lista com os nós desejados.
2. Para esse trabalho, modifique o código do trabalho 3 de forma a permitir uma análise nodal modificada, no tempo, de um circuito que pode conter componentes não-lineares (diodos). **A análise no tempo deve ser feita utilizando o método de Backward Euler e o diodo deve ser simulado considerando o método de Newton-Raphson e o modelo exponencial.**
3. O trabalho pode ser escrito em um ou mais arquivos “.py” 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 os seis argumentos descritos acima: uma string com o nome de um arquivo de texto que contém a “netlist”, um float que representa o tempo de simulação em segundos, um float que representa o passo, um float que representa a tolerância do Newton-Raphson (epsilon), uma lista de valores iniciais para cada nó da netlist, e uma lista com os nós desejados (isto é, a função deve ser definida da seguinte forma: `def main(arqNetlist, tSim, passo, tol, v0, vout)`). Ao fazer a chamada da função dentro de um “.py” o nome do arquivo deve ser colocado como uma string, ex.: `main('netlist1.txt', 1e-3, 10e-6, [0,1,0], [2])`. Nesse exemplo, o nome do arquivo da netlist é netlist1.txt, o tempo máximo de simulação é de 1 ms, o passo é de 10  $\mu$ s, a tensão inicial do nó 1 é 0, a tensão inicial do nó 2 é 1, e a tensão inicial do nó 3 é 0, e desejamos que a saída seja um array do numpy com as tensões do nó 2 de zero segundos a tSim segundos. A função main deve calcular e retornar um array do numpy, onde cada linha do array é um nó desejado, e cada coluna é referente ao valor daquele nó em cada instante de tempo. Por exemplo, se a entrada for (`'netlist1.txt', 1e-3, 0.2e-3, [0,1,0], [1,2]`), a saída deve ser um array no formato  $2 \times 6$ , onde a primeira linha contém os valores da tensão no nó 1 nos instantes de tempo 0 ms, 0.2 ms, 0.4 ms, 0.6 ms, 0.8 ms e 1 ms; e a segunda linha contém os valores da tensão no nó 2 nos mesmos seis instantes de tempo.

Caso o passo não resulte em uma divisão inteira em relação ao  $t_{sim}$ , considere o tempo de parada como sendo aquele imediatamente anterior a  $t_{sim}$ . Por exemplo, se a entrada for ('netlist1.txt',  $1e-3$ ,  $0.3e-3$ , [0,1,0], [3]), a saída deve ser um array no formato  $1 \times 4$ , onde a linha contém os valores da tensão no nó 3 nos instantes de tempo 0 ms, 0.3 ms, 0.6 ms, 0.9 ms.

4. O array retornado pela função main deve conter uma quantidade de linhas igual ao tamanho do vetor do último argumento, e uma quantidade de colunas igual à quantidade de pontos no tempo que são definidos de acordo com o tempo de simulação e com o passo. Considere que o vetor de entrada sempre estará na ordem crescente.
5. O arquivo ".py" que contém a função main deve ter o nome seguindo o seguinte padrão: trab4nomesobrenome.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: trab4fernandaoliveira.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.

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 senoidais, capacitores, indutores e transformadores, 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 controlada por corrente, considerando que a corrente de controle passa por um curto do nó C ao nó D: F<identificação> <nóI(fonte drena a corrente desse nó)> <nóI(fonte injeta a corrente nesse nó)> <nó de controle C> <nó de controle D> <valor do ganho de corrente B>
- Fonte de tensão controlada por tensão: E<identificação> <nó positivo> <nó negativo> <nó de controle positivo> <nó de controle negativo> <valor do ganho de tensão A>
- Fonte de tensão controlada por corrente, considerando que a corrente de controle passa por um curto do nó C ao nó D: H<identificação> <nó positivo> <nó negativo> <nó de controle C> <nó de controle D> <valor da transresistência Rm>
- Fonte de corrente: I<identificação> <nó cuja corrente é drenada pela fonte> <nó cuja corrente é injetada pela fonte> <parâmetros>
- Fonte de tensão: V<identificação> <nó positivo> <nó negativo> <parâmetros>

- Tanto para a fonte de corrente, quanto a fonte de tensão, os parâmetros podem ser:
  - Fonte constante: DC <valor>
  - Fonte senoidal, considerando a expressão  $i(t) = \text{<amplitude>} * \cos(2*\pi*\text{<frequência>}*t + \text{<fase>}*\pi/180)$  (ou  $v(t) = \text{<amplitude>} * \cos(2*\pi*\text{<frequência>}*t + \text{<fase>}*\pi/180)$ ) quando os parâmetros forem: SIN <valor DC> <amplitude> <frequência em Hz> <fase em graus>
  - Fonte pulse: PULSE <V1> <V2> <delay> <Trise> <Tfall> <Tv2> <Tperiod>, onde V1 é o primeiro nível de tensão; V2 é o segundo nível de tensão; delay é o tempo inicial, em segundos, com valor V1; Trise é o tempo de subida em segundos; Tfall é o tempo de descida em segundos; Tv2 é o tempo em V2; Tperiod é o período, em segundos. Se o tempo de delay for igual a 0, o sinal começa subindo (ou descendo) de V1 para V2. Se for diferente de zero, o sinal começa constante em V1.
- Indutor: L<identificação> <nó1> <nó2> <valor da indutância> <condição inicial, considerando o sentido do nó 1 para o nó 2>
- Capacitor: C<identificação> <nó1> <nó2> <valor da capacitância> <condição inicial, considerando o nó 1 como positivo>
- Transformador: K<identificação> <nó a> <nó b> <nó c> <nó d> <valor da indutância no primeiro enrolamento> <valor da indutância no segundo enrolamento> <valor da indutância mútua>
- Diodo: D<identificação> <nó positivo> <nó negativo> <I<sub>s</sub>> <ηV<sub>T</sub>>

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