

## Circuitos Elétricos II – 1º semestre de 2007 - Trabalho

Fazer um programa que analise circuitos no tempo, e que aceite, no mínimo, os elementos:

- Fontes de corrente e de tensão independentes (DC, pulso, senóide).
- Resistores, capacitores e indutores.
- As quatro fontes controladas.
- Diodos.
- Transistores bipolares NPN e PNP.

O programa deve ler um netlist, e realizar uma análise transiente, com parâmetros dados pelo operador (tempo, passo fixo). O resultado deverá ser uma tabela em arquivo, tendo o tempo como primeira coluna, e todas as tensões nodais e correntes nas fontes de tensão nas outras colunas, plotável com outro programa (XYplot, Gnuplot).

Capacitores e indutores devem poder ter uma condição inicial.

Os valores numéricos devem ser números (sem multiplicadores literais, como n, u, m u, k, etc). Também não é necessário suportar nomes de nós. Pode-se assumir os nós numerados seqüencialmente a partir de 0 (terra) e dados como números.

Os diodos devem usar o modelo  $i = I_s (e^{v/V_t} - 1)$ , com extensão reta se  $v/V_t > 36$ , com  $I_s$  e  $V_t$  dados opcionalmente. Os valores “default” devem resultar em um diodo que conduza 1 mA com  $v = 0.6$  V, e  $V_t = 25$  mV ( $I_s = 3.7751345e-14$ ).

Os transistores devem ser modelados pelo modelo de Ebers-Moll, com os parâmetros  $\alpha$  e  $\alpha_r$  dados. Os parâmetros  $I_s$  e  $V_t$  dos diodos do transistor devem poder também ser dados, opcionalmente, com os valores “default” usados se não forem dados.

O programa deve poder usar o método “backward” de Euler ou o método de segunda ordem de Gear, acoplado ao método de Newton-Raphson, usando análise nodal modificada. O primeiro passo é sempre com o método BE. Uma primeira análise com passo pequeno (1/1000 do normal) deve ser feita para achar a solução completa em  $t=0$ . A parte básica da MNA em C pode ser vista no programa <http://www.coe.ufrj.br/~acmq/cursos/mna1.zip>

Formato para o netlist:

Primeira linha: Número de nós (ou considerar comentário, se o programa contar os nós)

Linhas seguintes: Descrição do circuito, com um elemento por linha.

Resistor: R<nome> <nó1> <nó2> <Resistência>

Indutor: L<nome> <nó1> <nó2> <Indutância> [IC=<corrente inicial>]

Capacitor: C<nome> <nó1> <nó2> <Capacitância> [IC=<tensão inicial>]

Fonte de tensão controlada a tensão: E<nome> <nóV+> <nóV-> <nóv+> <nóv-> <Av>

Fonte de corrente controlada a corrente: F<nome> <nóI+> <nóI-> <nói+> <nói-> <Ai>

Fonte de corrente controlada a tensão: G<nome> <nóI+> <nóI-> <nóv+> <nóv-> <Gm>

Fonte de tensão controlada a corrente: H<nome> <nóV+> <nóV-> <nói+> <nói-> <Rm>

Fonte de corrente: I<nome> <nó+> <nó-> <Parâmetros>

Fonte de tensão: V<nome> <nó+> <nó-> <Parâmetros>

Diodo: D<nome> <nó anodo> <nó catodo> [<I<sub>s</sub>> <V<sub>t</sub>>]

Transistor: Q<nome> <nó c> <nó b> <nó e> NPN|PNP <α> <α<sub>r</sub>> [<I<sub>sbe</sub>> <V<sub>tbe</sub>> <I<sub>sbc</sub>> <V<sub>tbc</sub>>]

Comentário: \*<comentário>

Os parâmetros para as fontes devem ser:

DC <valor> (para fonte DC)

SIN (<nível contínuo> <amplitude> <frequencia (Hz)> <atraso> <fator de atenuação> <ângulo> <número de ciclos>)

PULSE (<amplitude 1> <amplitude 2> <atraso> <tempo de subida> <tempo de descida> <tempo ligada> <período> <número de ciclos>)

A fonte começa na amplitude 1, e fica aí até o fim do tempo de atraso. Então muda para a amplitude 2 variando linearmente dentro do tempo de subida, fica na amplitude 2 durante o tempo ligada, volta à amplitude 1 dentro do tempo de descida, e repete tudo com o período e o número de ciclos especificados. Termina na amplitude 1.

A fonte senoidal vale ( $X_i$  = parâmetro #i):

Antes do tempo de atraso  $X_4$  ou depois do numero de ciclos  $X_7$ :

Em outros tempos:

As direções para fontes e condições iniciais são de acordo com a ordem dos nós e as direções convencionais associadas, sendo o primeiro nó o positivo.

O programa deve ler as instruções de como tratar o netlist de uma linha de comando no próprio netlist, no formato:

```
.tran <tempo final> <passo> <BE|GEAR> <passos internos>
```

Os passos internos servem para aumentar a precisão sem ter que gerar tabelas muito grandes. Entre cada passo listado na tabela é feito o número especificado de passos iguais, com resultados não listados.

O programa deve contar quantas vezes o ciclo de Newton-Raphson é executado, e se o número passar de um valor razoável, tentar outra aproximação inicial para a solução. Deve contar também quantas vezes faz isto, e se o número passar de um valor razoável, abortar a análise. Em qualquer caso, deve haver meio do usuário interromper a análise.

Opções:

Inclua o amp. op. ideal e indutores acoplados, Implemente também o método dos trapézios. Plote gráficos. Use passo variável. Use um método mais sofisticado para evitar problemas de convergência.

O programa deve ser escrito em uma linguagem compilada como C, C++ ou Pascal. Não é aceitável usar Matlab ou similar. O programa deve rodar em ambiente Windows. Um arquivo .zip com tudo o que for necessário para o programa, inclusive fontes e arquivo executável, não deve ter mais de 1 Mbyte. O programa fonte deve consistir do mínimo número de arquivos permitido no ambiente de desenvolvimento escolhido.

Grupos de 3 alunos, no máximo. O programa deverá ser apresentado e demonstrado por todo o grupo, e entregue com um relatório com comentários e exemplos significativos e originais verificados. Trabalhos não originais ou não funcionando corretamente não serão considerados.