

## Circuitos Elétricos II – 2<sup>o</sup> semestre de 2010 - 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.
- Chaves controladas a tensão e resistores não lineares por partes.

O programa deve ler um “netlist”, e realizar uma análise transiente com passo de tempo variável, com parâmetros dados por um comando no netlist. 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, separadas por espaços, plotável com outro programa.

O programa deve usar o método “backward” de Euler, acoplado ao método de Newton-Raphson, usando análise nodal modificada. Uma primeira análise de ponto de operação deve ser feita para achar a solução em  $t=0$ , com capacitores substituídos por resistores de  $10^9 \Omega$  e indutores substituídos por resistores de  $10^{-9} \Omega$ . Condições iniciais não são consideradas. Devem ser colocadas explicitamente como fontes junto a capacitores e indutores, se necessário.

O programa básico de análise nodal modificada em C MNA1, pode ser usado como base:

<http://www.coe.ufrj.br/~acmq/cursos/mna1.zip>

Formato para o “netlist”:

Primeira linha: É um título, ignorado.

Linhas seguintes: Descrição do circuito, com um elemento por linha. O formato é livre, com um número qualquer de espaços separando os itens.

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

Indutor: **L**<nome> <nó1> <nó2> <Indutância>

Capacitor: **C**<nome> <nó1> <nó2> <Capacitância>

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>

Chave: **S**<nome> <no+> <no-> <noctrl+> <noctrl-> <gon> <goff> <vref>

Resistor não linear por partes: **N**<nome> <nó+> <nó-> <4 pontos vi ji >

Comentário: \*<comentário>

Os valores numéricos devem ser números (sem multiplicadores literais, como n, u, m u, k, etc).

As fontes F e H tem um ramo de entrada em curto. O formato é diferente do usual do Spice.

A chave tem especificadas as condutâncias quando ligada (gon) e desligada (goff), e a tensão de controle que a liga.

O resistor não linear segue a curva (em três segmentos de reta) determinada pelos quatro pontos dados. Para tensões fora da faixa, segue as retas das extremidades.

Os netlists de entrada podem ser gerados diretamente com o programa editor Edfil (o que também evita que o programa troque os números dos nós, se for usado o algoritmo de numeração do programa exemplo), disponível em <http://www.coe.ufrj.br/~acmq/programs>

Os parâmetros para as fontes de tensão ou de corrente 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 número de ciclos  $X_7$ :

$$X_1 + X_2 \operatorname{sen}\left(\pi \frac{X_3}{180}\right)$$

Em outros tempos:

$$X_1 + X_2 e^{(t-X_4)X_5} \operatorname{sen}\left(2\pi X_3(t - X_4) + \pi \frac{X_6}{180}\right)$$

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 fazer a análise de uma linha de comando no próprio netlist, em qualquer lugar, no formato:

**.tran** <tempo final> <passo mínimo> <passo máximo>

O programa deve começar a análise usando o passo mínimo, e ao fim de cada passo de tempo executar um algoritmo que procure otimizar o passo, respeitando os limites dados, a ser escolhido e testado no trabalho.

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. É interessante permitir que o usuário interrompa a análise, se estiver demorando demais.

O programa deve gerar uma tabela em arquivo de texto, contendo na primeira linha os nomes das variáveis (tempo="t", correntes ="j<nome do elemento>") e nas linhas seguintes os valores calculados com o tempo na primeira coluna.

O programa "mnad" pode ser usado para plotar os resultados e verificar análises. Na versão atual ele faz a análise pedida com passo fixo, que deve ser idêntica ao caso em que os passos mínimo e máximo são iguais. O programa será atualizado para fazer como pedido neste trabalho em breve.

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 **2 Mbytes**. 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, que demonstrem que o programa funciona corretamente. Trabalhos não originais ou não funcionando corretamente de acordo com o especificado não serão considerados.