

Circuitos Elétricos II – 2º semestre de 2011 - Trabalho

Prof. Antônio Carlos Moreirão de Queiroz

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).
- capacitores e indutores lineares invariantes no tempo.
- Resistores, possivelmente lineares por partes,
- As quatro fontes controladas, possivelmente lineares por partes.
- Amplificadores operacionais ideais, de 4 terminais.

O programa deve ler um netlist, e realizar uma análise transiente, com parâmetros dados por uma linha de 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, plotável com outro programa. A primeira linha da tabela deve listar os nomes de todas as variáveis calculadas, com correntes citadas como “j” seguidas do nome do elemento onde estão. Ex: t 1 2 3 4 5 6 jH1 jF2.

O método de integração numérica a usar é o dos trapézios. Inicialmente deverá ser feita uma análise de ponto de operação para a determinação das condições iniciais, com capacitores em aberto e indutores em curto-circuito. (Melhor colocar resistores grandes e pequenos nos lugares de capacitores e indutores.)

A análise deverá ser feita usando análise nodal, com elementos contendo fontes de tensão modelados com modelos baseados em amplificadores operacionais, de forma a nunca aumentar o tamanho do sistema de equações além do número de nós, descontando o de terra. O programa pode ser baseado no programa exemplo mna1amp, que implementa o algoritmo para um circuito resistivo linear:

<http://www.coe.ufrj.br/~acmq/cursos/mna1amp.zip>

Formato para o netlist:

O netlist pode ser gerado pelo programa Edfil, e deve ser compatível com o aceito pelo programa exemplo mnad. Veja os programas em <http://www.coe.ufrj.br/~acmq/cursos>

Primeira linha: Comentário, ignorar (O editor Edfil coloca o número de nós nesta linha).

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

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>

Amplificador operacional ideal: O<nome> <nó saída+> <nó saída-> <nó entrada+> <nó entrada->

Comentário: *<comentário>

O programa exemplo permite nomes nos nós. O programa feito pode continuar permitindo usando o mesmo algoritmo, ou admitir apenas números. Neste caso a primeira linha gerada pelo editor Edfil pode ser usada.

As direções para fontes são de acordo com a ordem dos nós e as direções convencionais associadas, sendo o primeiro nó o positivo. Os parâmetros para as fontes podem ser:

DC <valor>

SIN <nível contínuo> <amplitude> <frequencia (Hz)> <atraso> <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 senoidal, antes do atraso ou após o número de ciclos vale: nível contínuo+amplitude*sen(π *ângulo/180). No resto do tempo vale: nível contínuo+amplitude*exp(-t0*atenuação)*sen($2*\pi$ *(frequência*t0+ângulo/360)), onde t0=tempo-atraso.

A fonte pulsada 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.

Para os elementos R, E, F, G e H, em vez do valor pode estar uma lista de 4 pares de valores {x, f(x)}, definindo uma curva linear por partes com 3 retas. Um diodo, por exemplo, poderia ser definido como um resistor com os parâmetros: -1000 -1e-6 0 0 0.6 1e-3 2 20. Veja o elemento "N" no programa mnad, que é um resistor assim. É recomendado fazer o programa aceitar também a inicial "N" para o resistor, pois fica difícil saber qual a polaridade de um resistor comum no editor Edfil.

O programa deve ler as instruções de como tratar o netlist de uma linha de comando no próprio netlist, no formato abaixo. Não deve ser necessário fornecer qualquer outro parâmetro ao programa além do arquivo de entrada. Os passos internos permitem aumentar a precisão da análise, com alguns passos entre os valores que vão à tabela de saída. O passo interno usado é então o passo dado dividido pelo número de passos internos.

.TRAN <passo> <tempo final> TRAP <passos internos>

O programa mnad pode ser usado para plotar os gráficos de saída e para verificação. Ele atualmente faz uma análise compatível com a pedida, inclusive com os elementos não lineares como especificado, mas usa análise nodal modificada normal.

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.

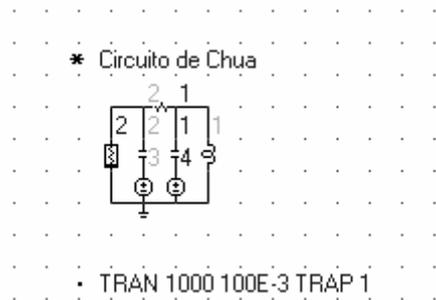
O programa deve ser escrito em uma linguagem compilada como C, C++ ou Pascal. O programa deve rodar em ambiente gráfico Windows. Um arquivo .zip com tudo o que for necessário para o programa, inclusive fontes, arquivo executável, documentação e exemplos não deve ter mais de 3 Mbytes. O programa fonte 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, até uma semana antes da segunda prova.

Alguns exemplos:

Circuito de Chua, oscilador caótico, simulado no programa mnad. Mude o elemento N para R e o programa do trabalho deve dar o mesmo resultado:

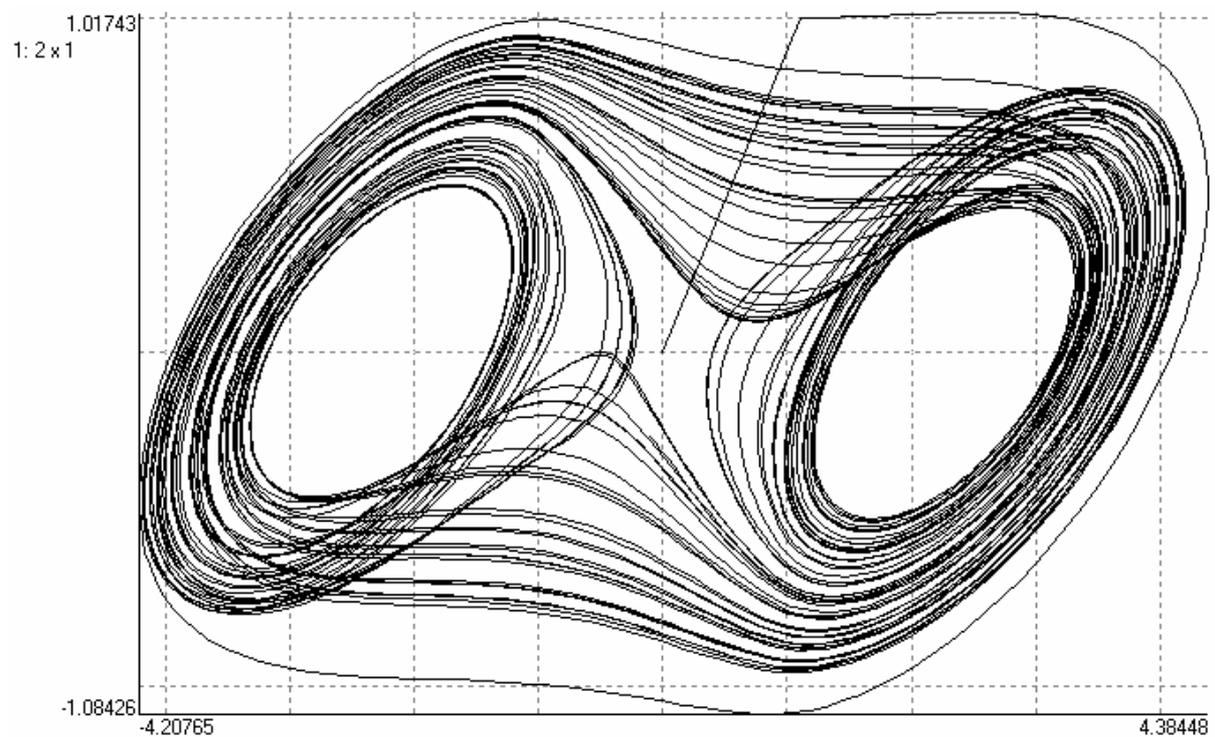
Esquemático no Edfil:



Netlist gerado:

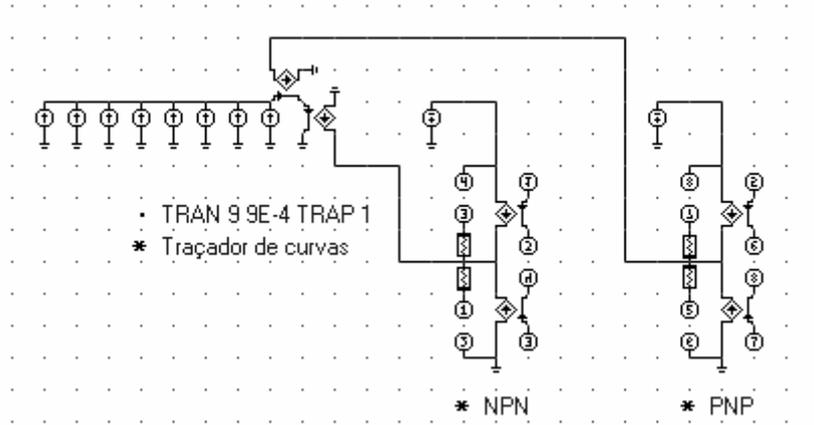
```
4
R0102 1 2 1.9
L0100 1 0 1
C0203 2 3 0.31
C0104 1 4 1
N0200 2 0 -2 1.1 -1 0.7 1 -0.7 2 -1.1
V0300 3 0 PULSE 0 1 0 0 0 10000 20000 1
V0400 4 0 PULSE 0 1 0 0 0 10000 20000 1
.TRAN 1000 100E-3 TRAP 1
```

Simulação no mnad:



Traçador de curvas de transistores bipolares:

As fontes fazem um traçador de curvas, e transistores bipolares NPN e PNP são simulados com o modelo de Ebers-Moll e diodos lineares por partes N. Transdutores lineares por partes podem ser usados no lugar dos amplificadores de corrente, controlados pelas tensões sobre os diodos com as mesmas curvas multiplicadas pelos alfas, dando o mesmo resultado.



```

10
N0102 1 2 -1000 0 0 0 0.6 1e-3 2 10
N0304 3 4 -1000 0 0 0 0.6 1e-3 2 10
N0504 5 4 -1000 0 0 0 0.6 1e-3 2 10
N0106 1 6 -1000 0 0 0 0.6 1e-3 2 10
F0001 0 1 6 7 0.5
F0004 0 4 5 8 0.5
F0701 7 1 2 0 0.99
F0804 8 4 3 0 0.99
F0001 0 1 9 0 0.01
F0400 4 0 10 9 0.01
I0010 0 10 PULSE 0 1 8 0 0 10 20 1
I0010 0 10 PULSE 0 1 7 0 0 10 20 1
I0010 0 10 PULSE 0 1 6 0 0 10 20 1
I0010 0 10 PULSE 0 1 5 0 0 10 20 1
I0010 0 10 PULSE 0 1 4 0 0 10 20 1
I0010 0 10 PULSE 0 1 3 0 0 10 20 1
I0010 0 10 PULSE 0 1 2 0 0 10 20 1
I0010 0 10 PULSE 0 1 1 0 0 10 20 1
V0007 0 7 PULSE 0 -10 0 0.5 0.5 0 1 9
V0008 0 8 PULSE 0 10 0 0.5 0.5 0 1 9
.TRAN 9 9E-4 TRAP 1
    
```

