Circuitos Elétricos II – 1º semestre de 2012 - Trabalho

Prof. Antônio Carlos Moreirão de Queiroz

Fazer um programa que analise circuitos no tempo para estudar os métodos de Gear, 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, lineares.
- 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 de Gear, com ordem especificada entre 1 e 8. Inicialmente deverá ser feita uma análise usando o método de ordem 1 e um passo muito menor que o normal, para completar a solução em t=0 a partir das condições iniciais sobre indutores e capacitores, e dos valores em t=0 das fontes independentes. A partir daí o método de Gear é usado. Valores de variáveis em t<0 requeridos pelo método são considerados iguais aos valores em t=0.

A análise deverá ser feita usando análise nodal modificada. O programa pode ser baseado no programa exemplo mna1, que implementa o algorítmo para um circuito resistivo linear:
http://www.coe.ufrj.br/~acmq/cursos/mna1.zip

Formato para o netlist:
O netlist pode ser gerado pelo programa Edfil, e deve ser compatível com o aceito pelo programa exemplo mnae. 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>
Resistor não linear> N<nome> <nó1> <nó2> <quatro pares de valores vi ji>
Chave controlada a tensão: $<nome> <no+> <no-> <noctrl+> <noctrl-> [<gon> <goff> <vref>]
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>
Amplificador operacional ideal: O<nome> <nó saída+> <nó saída-> <nó entrada+> <nó entrada->
Comentário: *<comentário>

Os programas exemplo permitem nomes nos nós. O programa feito pode continuar permitindo usando o mesmo algorítmo, 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>

Atualizado em 28/6/2012

SIN <nivel contínuo> <amplitude> <frequência (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(ângulo*π/180). No resto do tempo vale: nível contínuo+amplitude*exp(-t0*atenuação)*sen(2*π*frequência*t0+ângulo*π/180), 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.
Cuidado com como tratar tempos de subida e de descida nulos. Pode ser usado o tempo do passo.

O resistor não linear "N" é definido por quatro pares de valores {*v*, *j*}, 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.

A chave é um resistor linear (um condutor na verdade), com valor da condutância dependendo da tensão de controle, valendo gon se vctrl ≥ vref e goff se vctrl<vref. Notar que o valor varia dentro da análise de Newton-Raphson.

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 <tempo final> <passo> GEAR[<n>] <passos internos> UIC
onde <n> é a ordem do método de Gear a usar. Se nada for dito é o de segunda ordem. O comando UIC significa "use initial conditions", pois normalmente programas de análise no tempo fazem uma análise de ponto de operação e acham dela as condições iniciais antes de iniciar a análise.

O programa mnae pode ser usado para plotar os gráficos de saída e para verificação (na versão atual ele vai sempre usar o método de Gear de ordem 2).

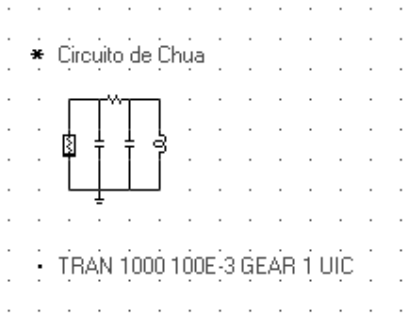
O programa deve contar quantas vezes o ciclo de Newton-Raphson é executado, e se o número passar de um valor razoável, tentar outra aproximação inicial para a solução. Deve contar também quantas vezes faz isto, e se o número passar de um valor razoável, abortar a análise.

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:

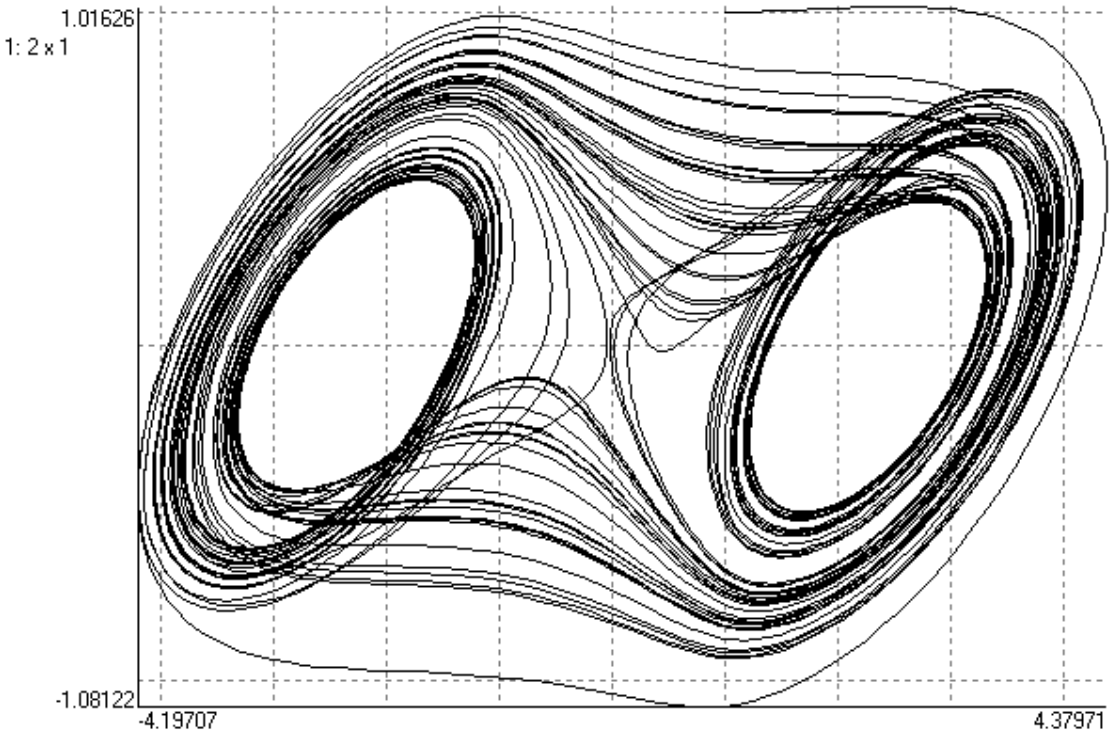
Circuito de Chua, oscilador caótico, simulado no programa mnae.
Esquemático no Edfil:



Netlist gerado pelo Edfil:

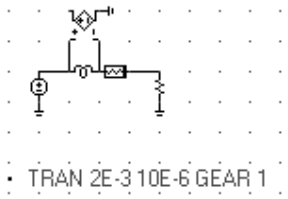
```
2
R0102 1 2 1.9
L0100 1 0 1
C0200 2 0 0.31 IC=1
C0100 1 0 1 IC=1
N0200 2 0 -2 1.1 -1 0.7 1 -0.7 2 -1.1
.TRAN 1000 100E-3 GEAR 1 UIC
```

Simulação no programa mnae:



Atualizado em 28/6/2012

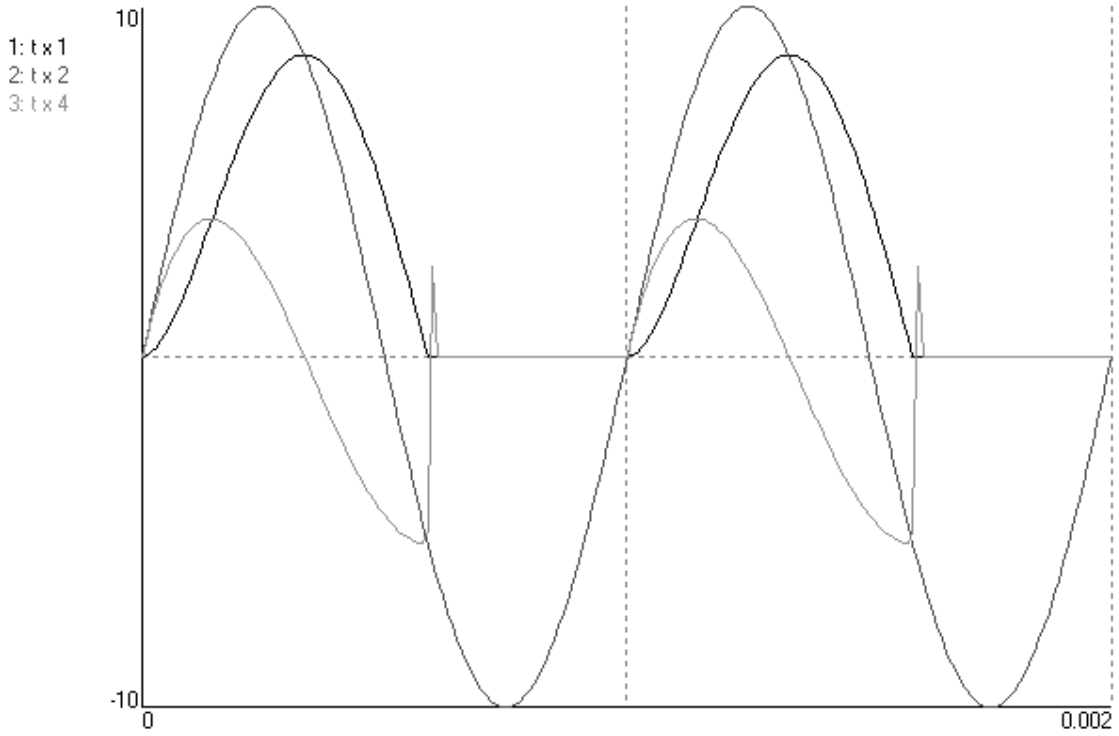
Retificador com filtro indutivo:



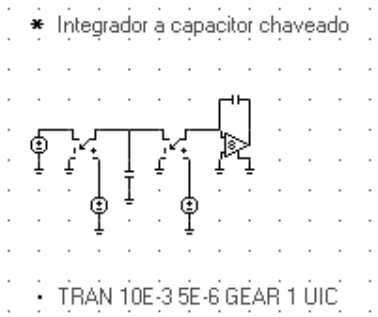
Netlist gerado pelo Edfil:

```
4
R0100 1 0 10
L0203 2 3 1E-3
N0301 3 1 -1000 0 0 0 0.1 1000 1 10000
E0400 4 0 2 3 1
V0200 2 0 SIN 0 10 1000 0 0 0 10
.TRAN 2E-3 10E-6 GEAR 1 UIC
```

Simulação no mnae. Notar o “dente” na tensão sobre o indutor.



Integrador a capacitor chaveado:

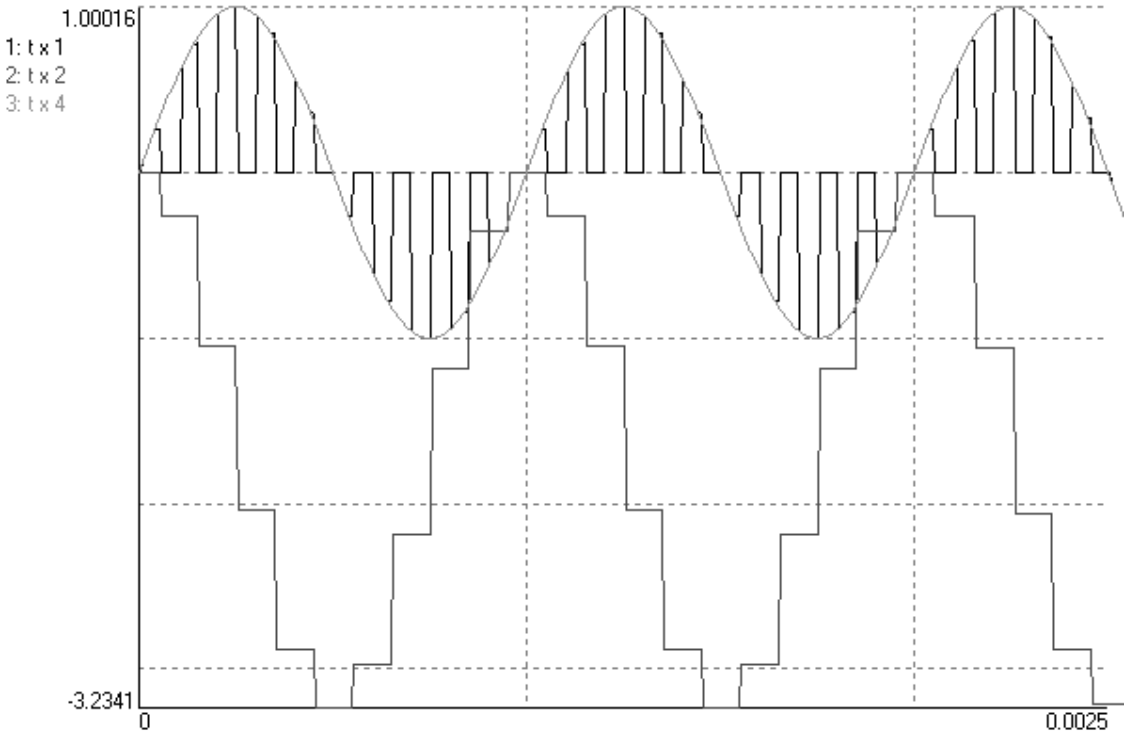


Netlist gerado pelo Edfil:

```
6
C0100 1 0 1E-9
C0203 2 3 1E-9
$0104 1 4 5 0 1 0 2.5
$0301 3 1 6 0 1 0 2.5
O0200 2 0 3 0
V0500 5 0 PULSE 0 5 0 0.01E-3 0.01E-3 0.03E-3 0.1E-3 10000
V0600 6 0 PULSE 0 5 0.05E-3 0.01E-3 0.01E-3 0.03E-3 0.1E-3 10000
V0400 4 0 SIN 0 1 1000 0 0 0 1000

.TRAN 10E-3 5E-6 GEAR 1 UIC
```

Resultado do mnae. Não aparecem artifatos:



Atualizado em 28/6/2012