

## Circuitos Elétricos II – 2º semestre de 2015 - Trabalho

Prof. Antônio Carlos Moreirão de Queiroz

Fazer um programa que analise circuitos no tempo para estudar os métodos implícitos de Adams-Moulton, e que aceite, no mínimo, os elementos:

- Fontes de corrente e de tensão independentes (DC, pulso, senoide).
- capacitores e indutores lineares invariantes no tempo.
- Resistores, possivelmente lineares por partes.
- As quatro fontes controladas, lineares.
- Chaves controladas a tensão.
- Amplificadores operacionais ideais, de 4 terminais.

O programa deve ler um “netlist”, e realizar uma análise transiente não linear, 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 Adams-Moulton, com ordem especificada entre 1 e 4. 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 Adams-Moulton é usado. Valores de variáveis em  $t < 0$  requeridos pelo método podem ser consideradas iguais aos valores em  $t = 0$ .

Os métodos de Adams-Moulton são implícitos, aproximando as funções a integrar como polinômios. Os dois primeiros são idênticos ao “backward” de Euler e o dos trapézios. Os modelos são como os nestes métodos.

$$y(t_0 + \Delta t) = y(t_0) + \int_{t_0}^{t_0 + \Delta t} f(t) dt$$

$$1) \quad y(t_0 + \Delta t) \approx y(t_0) + \Delta t f(t_0 + \Delta t)$$

$$2) \quad y(t_0 + \Delta t) \approx y(t_0) + \Delta t \left( \frac{1}{2} f(t_0) + \frac{1}{2} f(t_0 + \Delta t) \right)$$

$$3) \quad y(t_0 + \Delta t) \approx y(t_0) + \Delta t \left( -\frac{1}{12} f(t_0 - \Delta t) + \frac{8}{12} f(t_0) + \frac{5}{12} f(t_0 + \Delta t) \right)$$

$$4) \quad y(t_0 + \Delta t) \approx y(t_0) + \Delta t \left( \frac{1}{24} f(t_0 - 2\Delta t) - \frac{5}{24} f(t_0 - \Delta t) + \frac{19}{24} f(t_0) + \frac{3}{8} f(t_0 + \Delta t) \right)$$

A análise deverá ser feita usando análise nodal modificada. O programa pode ser baseado no programa exemplo MNA1, que implementa o algoritmo 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: Título, ignorar (O editor Edfil coloca o número de nós nesta linha).

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

Resistor: R<nome> <nó<sub>1</sub>> <nó<sub>2</sub>> <R>

Resistor não linear: N<nome> <nó<sub>1</sub>> <nó<sub>2</sub>> <quatro pares de valores  $v_i, j_i$ >

Chave controlada a tensão: \$<nome> <no+> <no-> <nó<sub>ctrl+</sub>> <nó<sub>ctrl-</sub>> [<g<sub>on</sub>> <g<sub>off</sub>> <v<sub>ref</sub>>]  
 Indutor: L<nome> <nó<sub>1</sub>> <nó<sub>2</sub>> <Indutância> [IC=<corrente inicial>]  
 Capacitor: C<nome> <nó<sub>1</sub>> <nó<sub>2</sub>> <Capacitância> [IC=<tensão inicial>]  
 Fonte de tensão controlada a tensão: E<nome> <nó<sub>v+</sub>> <nó<sub>v-</sub>> <nó<sub>v+</sub>> <nó<sub>v-</sub>> <Av>  
 Fonte de corrente controlada a corrente: F<nome> <nó<sub>i+</sub>> <nó<sub>i-</sub>> <nó<sub>i+</sub>> <nó<sub>i-</sub>> <Ai>  
 Fonte de corrente controlada a tensão: G<nome> <nó<sub>i+</sub>> <nó<sub>i-</sub>> <nó<sub>v+</sub>> <nó<sub>v-</sub>> <Gm>  
 Fonte de tensão controlada a corrente: H<nome> <nó<sub>v+</sub>> <nó<sub>v-</sub>> <nó<sub>i+</sub>> <nó<sub>i-</sub>> <Rm>  
 Fonte de corrente: I<nome> <nó<sub>+</sub>> <nó<sub>-</sub>> <Parâmetros>  
 Fonte de tensão: V<nome> <nó<sub>+</sub>> <nó<sub>-</sub>> <Parâmetros>  
 Amplificador operacional ideal: O<nome> <nó<sub>saída+</sub>> <nó<sub>saída-</sub>> <nó<sub>entrada+</sub>> <nó<sub>entrada-</sub>>  
 Comentário: \*<comentário>

Os programas exemplo permitem 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> <frequência em Hz> <atraso> <amortecimento> <defasagem em graus>  
 <número de ciclos>  
 PULSE <amplitude 1> <amplitude 2> <atraso> <tempo de subida> <tempo de descida> <tempo ligada>  
 <período> <número de ciclos>

A fonte senoidal vale:

$$x(t) = A_0 + Ae^{-\alpha(t-t_a)} \sin\left(2\pi f(t-t_a) + \frac{\pi}{180} \varphi\right)$$

onde  $A_0$  é o nível contínuo,  $A$  a amplitude,  $f$  a frequência,  $t_a$  o atraso,  $\alpha$  o amortecimento e  $\varphi$  a defasagem. Antes de  $t = t_a$  ou após o número de ciclos, tem o valor inicial ou final respectivamente, de forma a não criar descontinuidades.

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. Os tempos de subida e de descida podem ser nulos. O programa pode usar o tempo do passo então.

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 condutor linear, com valor da condutância dependendo da tensão de controle, valendo  $g_{on}$  se  $v_{ctrl} \geq v_{ref}$  e  $g_{off}$  se  $v_{ctrl} < v_{ref}$ . 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> ADMO<n> <passos internos> UIC onde <n> é a ordem do método de Adams-Moulton a usar. 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 estão implementados apenas os métodos de ordens 1 e 2, que são o “backward” de Euler (BE) e o dos trapézios (TRAP). O programa deverá ser atualizado em breve).

O método requer o cálculo e armazenamento de tensões e correntes em capacitores e indutores. A análise nodal modificada calcula ambas para indutores mas só as tensões nos capacitores. **As correntes neles devem ser calculadas ao fim da análise** (É ineficiente tratar capacitores com cálculo de corrente.) Podem ser colocadas na tabela gerada também.

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++, Pascal, etc. O programa deve rodar em ambiente Windows (32 ou 64 bits). 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.

Grupos de 3 alunos, no máximo. O programa deverá ser apresentado e demonstrado por todo o grupo, e entregue com um relatório (em arquivo) com comentários e exemplos significativos e originais verificados, até duas semanas antes da segunda prova. O trabalho é complexo, e deve ser iniciado imediatamente.