

**Programa que analise circuitos lineares contendo transistores MOS, encontrando o ponto de operação e a resposta em frequência para pequenos sinais.**

Nome: Diogo Nocera Magalhães

Nome: Felipe Claudio da Silva Santos

Professor: Antônio Carlos Moreirão de Queiroz

**Comentários sobre o programa:**

O programa analisa circuitos presentes em arquivos .net, nos quais há a descrição da conexão de cada elemento do mesmo. Foram utilizados como base os programas MNA1 e MNA1GR. O programa gera como saída um arquivo com a resposta em frequência do esquemático analisado. O trabalho pode ser divido em duas partes principais:

**Ponto de operação**

**Capacitor:** É modelado como uma condutância de valor muito pequeno (1e-9) e é calculada a tensão sobre o mesmo.

**Indutor:** É modelado como um resistor de valor muito pequeno (1e-9) e calcula-se a corrente que passa pelo mesmo. Um nó a mais é criado para que esta análise possa ser realizada. Este modelamento é feito para facilitar a posterior análise “AC” do transformador.

**Transformador:** Como o mesmo é formado por dois indutores acoplados, as correntes “DC” sobre os mesmos já são analisadas previamente.

**Transistor MOS**: As capacitâncias (entre o gate e a base, gate e o dreno, gate e source) são transformadas em condutâncias de valor muito pequeno (1e-9). As condutâncias “gm”, “gmb” e “gds” também foram modeladas. A princípio o transistor NMOS foi modelado, e a partir dele foi o modelado o PMOS (invertendo a corrente) Alguns cuidados extra tiverem que ser tomados com relação o vbs (não pode ultrapassar phi/2) e a determinação dos terminais de dreno e source, já que o elemento é simétrico com relação à isto.

**Algoritmo de Convergência do circuito:** Foi utilizado o algoritmo de Newton-Raphson, para buscar o ponto de operação, e, assim, possibilitar a análise em frequência posterior. Cada elemento do vetor de resposta do circuito (tensões e correntes calculadas) é analisado com relação ao que foi calculado antes, buscando assim achar uma configuração em que o erro diminua a cada iteração do algoritmo. Caso uma das respostas não esteja convergindo, um valor randomizado é atribuído à mesma, tendo tirar a análise da uma região onde a resposta não converge.

**Analise na frequência**

Para este tipo de análise foi utilizada a biblioteca de números complexos do c++. O programa pode trabalhar frequências em Hz ou rad.

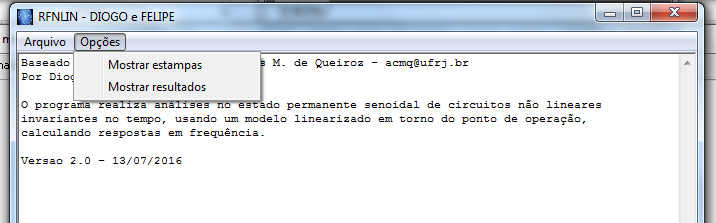
**Capacitores e Indutores:** estes elementos tem suas respectivas reatâncias varáveis com a frequência, sendo estes montados numa matriz complexa similar à encontrada no cálculo de ponto de operação.

**Transformador:** Para montar o transformador, é necessário descobrir quais linhas e colunas estão relacionadas aos acoplamentos dos indutores e adicionar o efeito do acoplamento.

**Transistor MOS:** As capacitâncias do transistor são tratadas como elementos uma capacitância normal, sendo estes os únicos componentes internos que vão variar com a frequência.

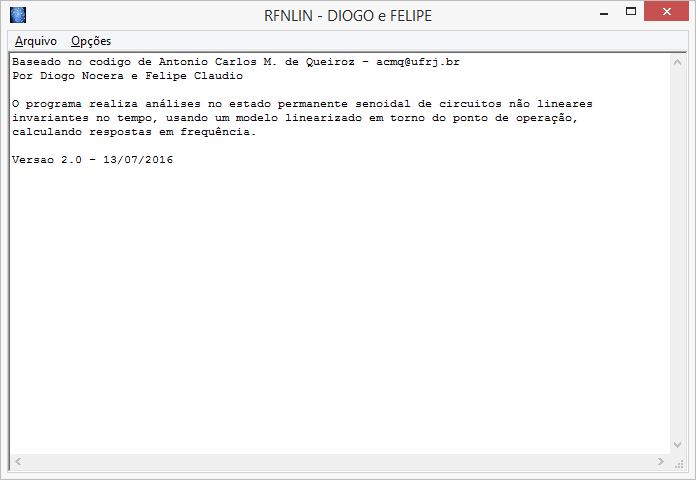
**Comentários sobre a interface**

A interface tem dois botões de opções:



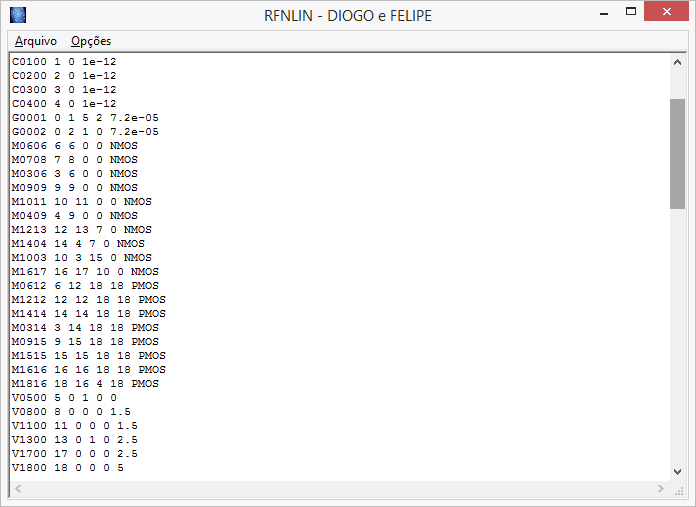
A primeira “Mostrar estampas”, mostra tudo que é feito pelo programa, inclusive os resultados do mesmo, a segunda “Mostrar resultados”, mostra somente os valores das variáveis após cada iteração do programa (útil para análise de circuitos muito grandes). Caso nenhum dos dois seja marcado, o programa só gera o .tab e não mostra o que foi calculado.

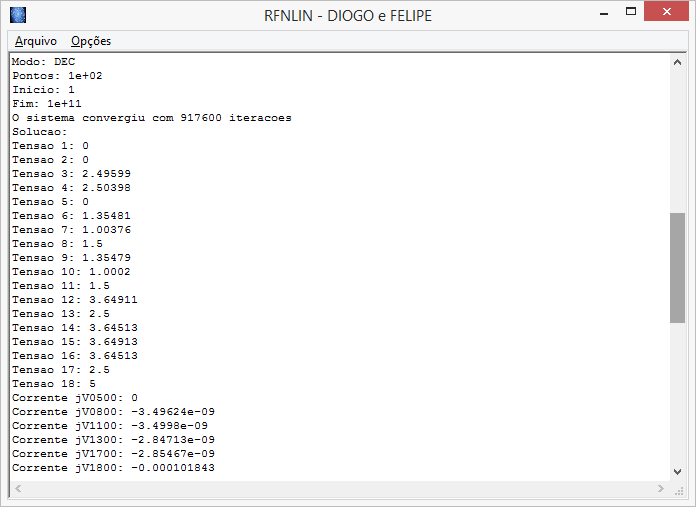
**INTERFACE**

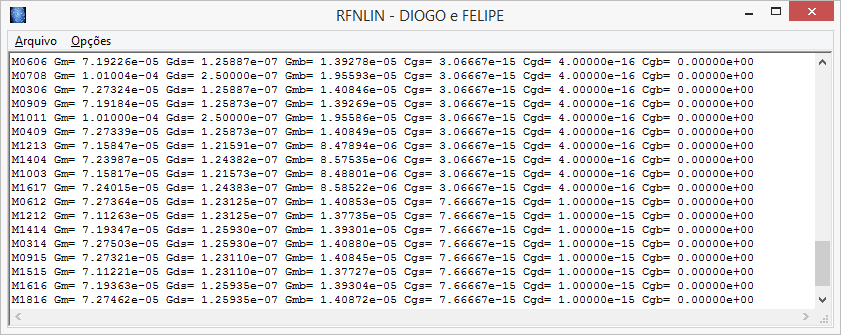
****

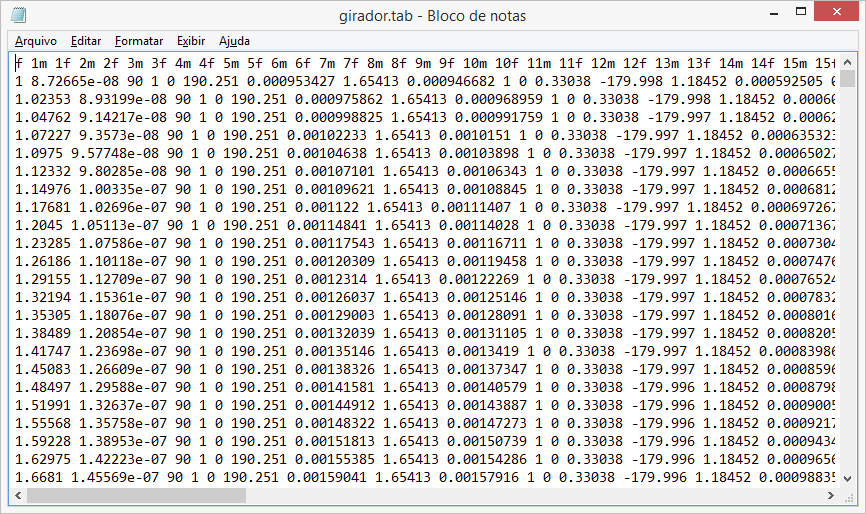
**Exemplos de operação**

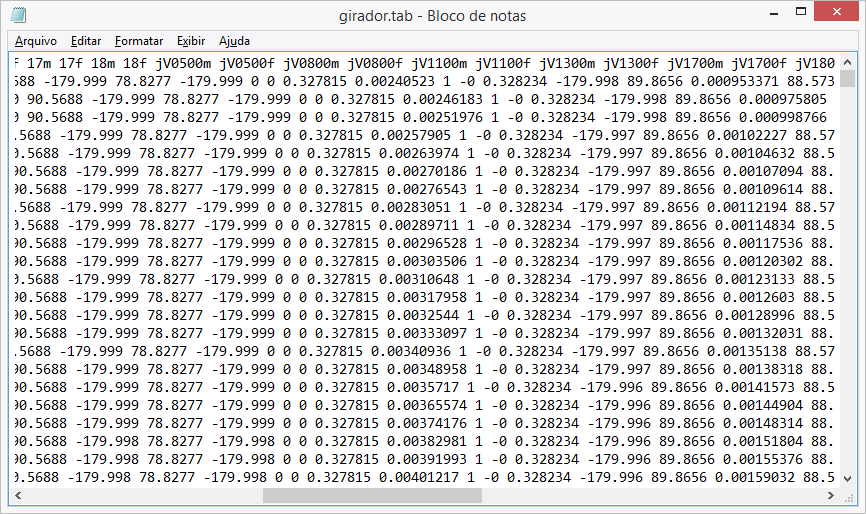
**GIRADOR.NET**

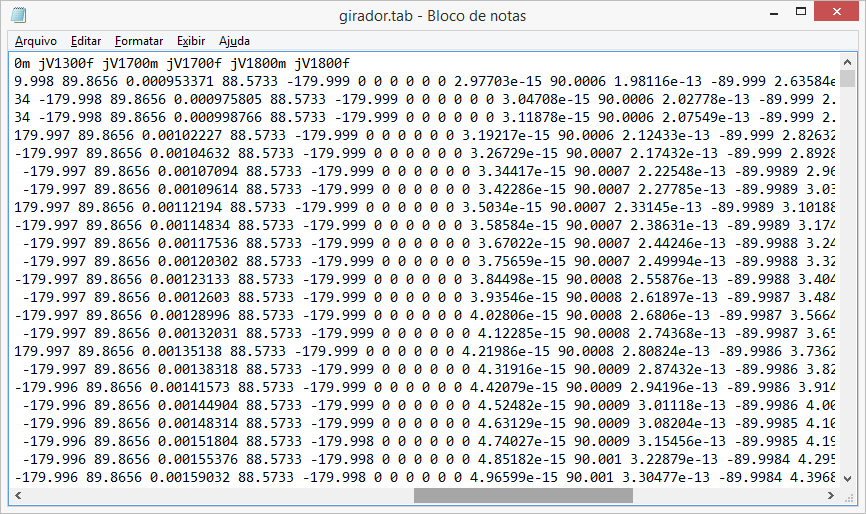
****

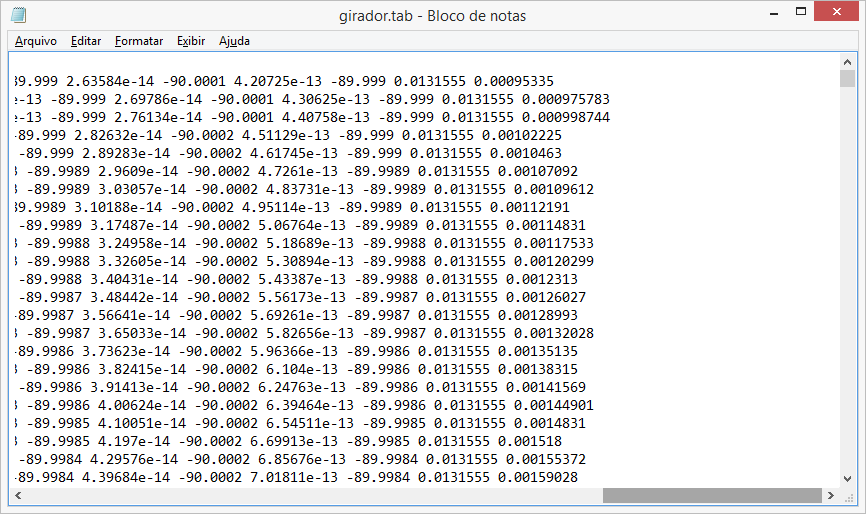
****

****

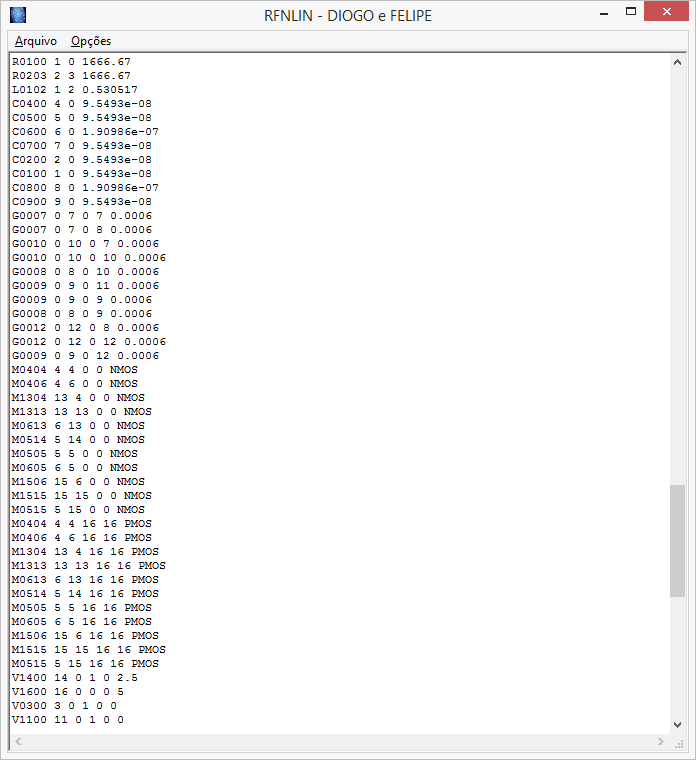
****

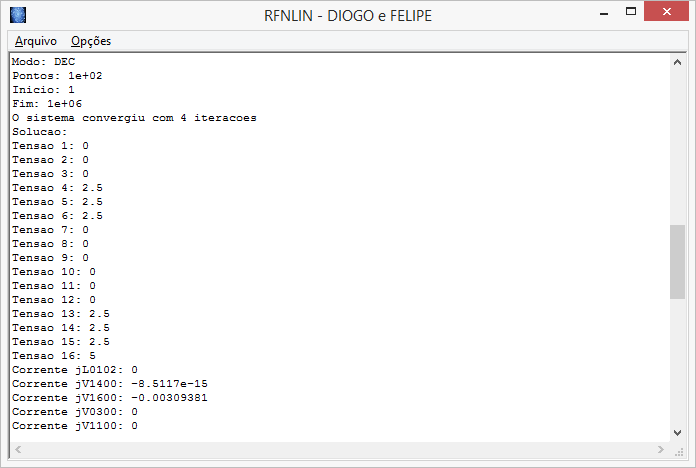
****

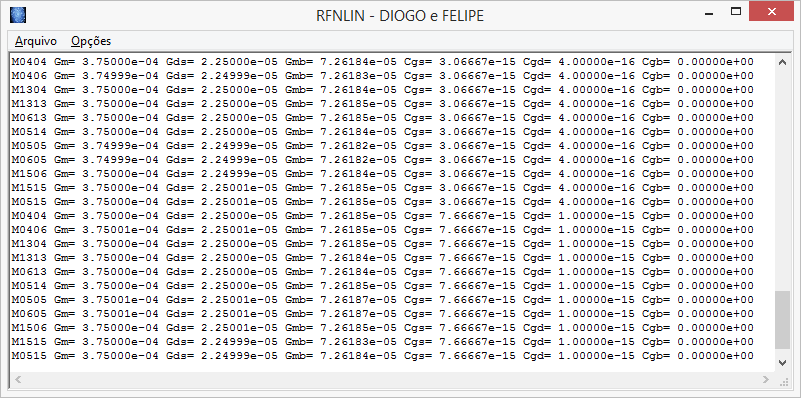
****

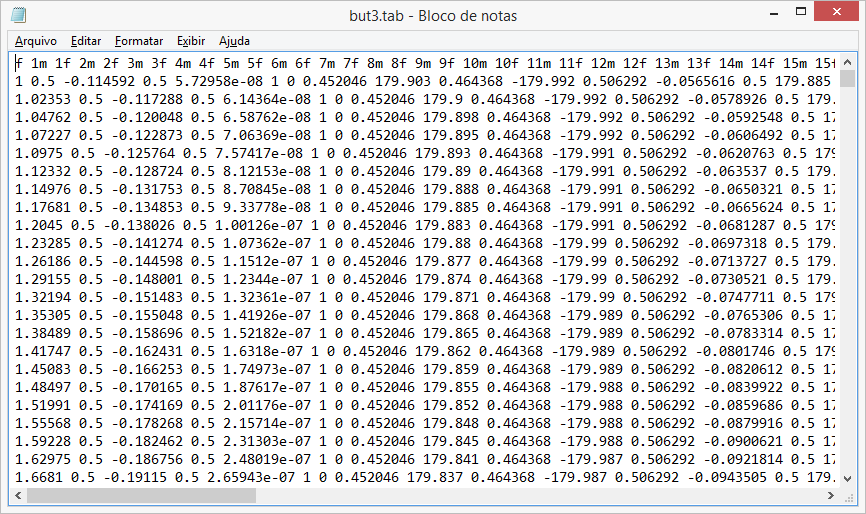
****

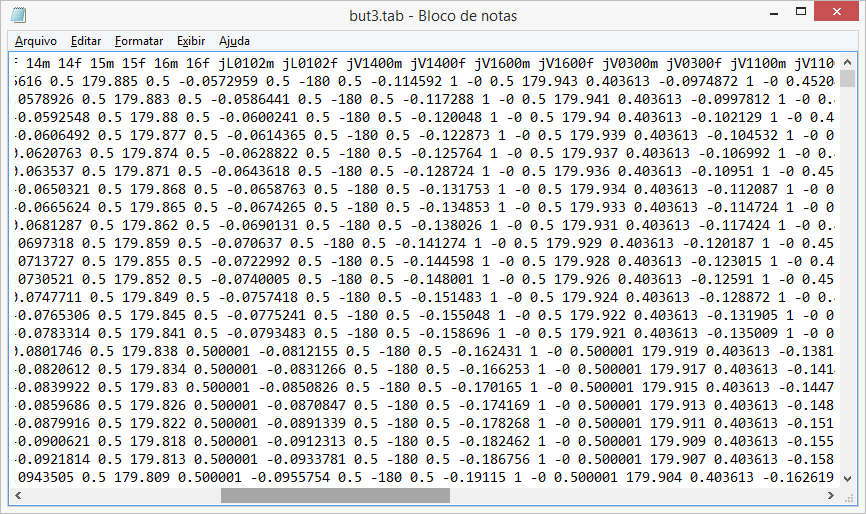
**BUT3.NET**

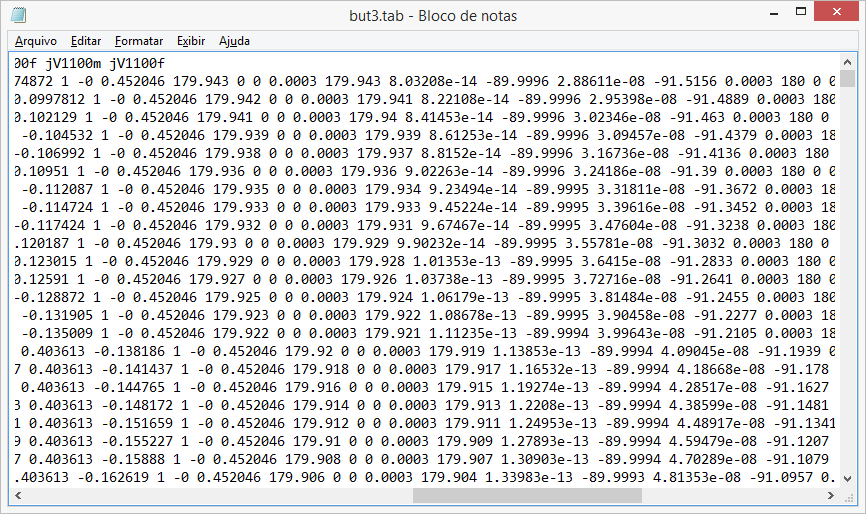
****

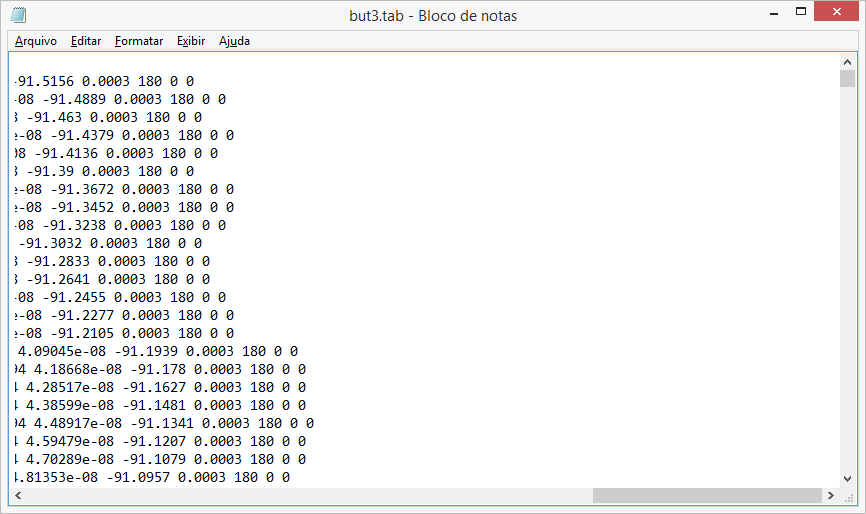
****

****

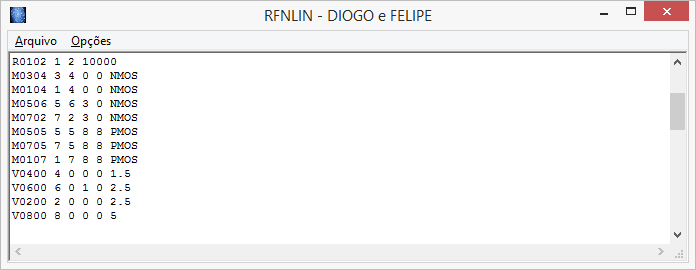
****

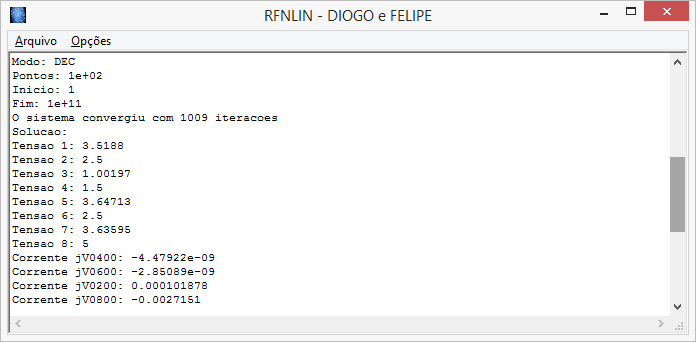
****

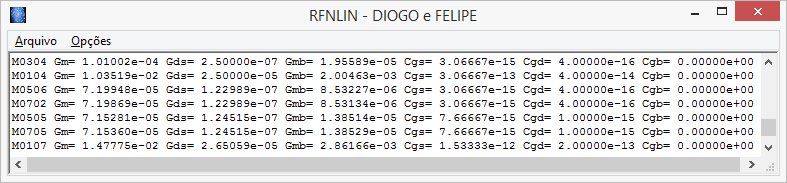
****

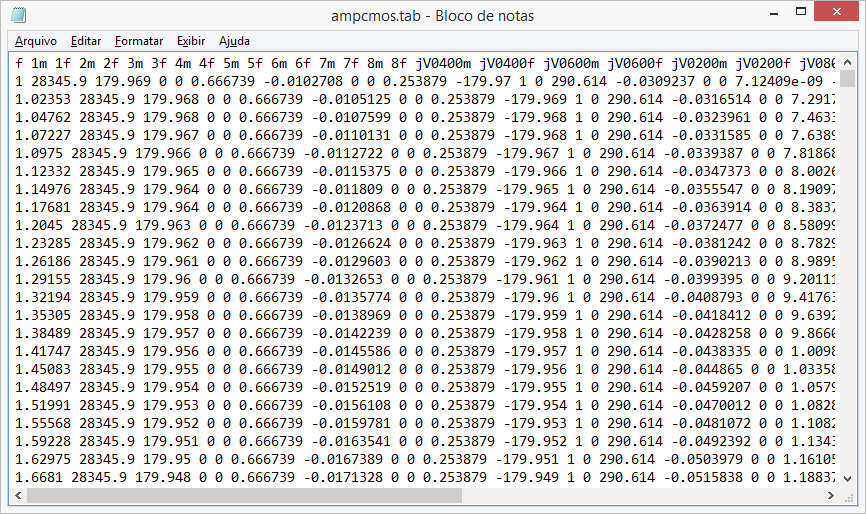
****

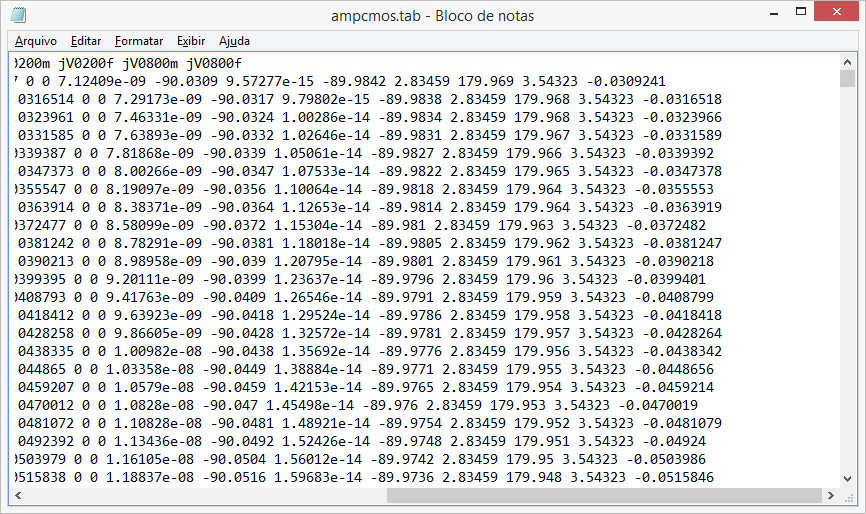
**AMPCMOS.NET**

****

****

****

****

****