



**UNIVERSIDADE FEDERAL DA BAHIA  
ESCOLA POLITÉCNICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**RICARDO ALMEIDA ARAUJO DA SILVA**

**OTIMIZAÇÃO DE PROJETOS DE CIRCUITOS INTEGRADOS ANALÓGICOS  
USANDO MÉTODOS DE PONTOS INTERIORES  
EM FUNÇÕES NÃO CONVEXAS E NÃO LINEARES  
NO AMBIENTE MATLAB**

Salvador  
2011

## Ficha Catalográfica

Silva, Ricardo Almeida Araujo da.

Otimização de projetos de circuitos integrados analógicos usando métodos de pontos interiores em funções não convexas e não lineares no ambiente Matlab./ Ricardo Almeida A. da Silva. Escola Politécnica. Programa de Pós-Graduação em Engenharia Elétrica, Salvador, 2011. f 146.

Orientadora: Profa. Dra. Ana Isabela Araújo Cunha.

Coorientadora: Profa. Dra. Luciana Martinez.

1.CMOS. 2. MOSFET 3. Integrador de Seevinck. 4.Circuitos Analógicos. 5. Otimização. 6. Restrições. 7. Programação Não Linear. 8. Funções Não Lineares. 9. Métodos dos Pontos Interiores. –

Engenharia I. Universidade Federal da Bahia. Escola Politécnica II. Cunha, Ana Isabela Araújo. III Título.

CDU:

**RICARDO ALMEIDA ARAUJO DA SILVA**

**OTIMIZAÇÃO DE PROJETOS DE CIRCUITOS INTEGRADOS ANALÓGICOS  
USANDO MÉTODOS DE PONTOS INTERIORES  
EM FUNÇÕES NÃO CONVEXAS E NÃO LINEARES  
NO AMBIENTE MATLAB**

Dissertação apresentada ao  
Programa de Pós-Graduação em  
Engenharia Elétrica,  
Escola Politécnica  
da  
Universidade Federal da Bahia,  
como requisito parcial para obtenção do  
grau de Mestre em Engenharia Elétrica  
e aprovada pela seguinte banca  
examinadora:

---

Profa. Dra. Ana Isabela Araújo Cunha  
Universidade Federal da Bahia

---

Profa. Dra. Luciana Martinez  
Universidade Federal da Bahia

---

Prof. Dr. Alessandro Girardi  
Universidade Federal do Pampa

---

Prof. Dr. Niraldo Roberto Ferreira  
Universidade Federal da Bahia

---

Prof. Dr. Amauri Oliveira  
Universidade Federal da Bahia

Salvador, 8 de julho de 2011

Para minha família.

## **AGRADECIMENTOS**

Aos mestres com quem tive a honra de conviver e o prazer de aprender:

Ana Isabela Araújo Cunha, Luciana Martinez, Carlos Eduardo Trabuco Dórea, Caiuby Alves da Costa, Adhemar de Barros Fontes, Antonio Cezar Castro Lima, Cristiane Correa Paim, Niraldo Roberto Ferreira, Amauri Oliveira, Evangivaldo Almeida Lima, Maximiliano Carvalho Santos, André Gustavo Scolari Conceição, Augusto Cesar Pinto Loureiro da Costa, além do prestimoso atendimento de Núbia Batista Araujo, Maria Janaína Batista, Nadia Fernanda e Jailton Moutinho. Aos colegas do Laboratório de Concepção de Circuitos Integrados - LCCI, pelo apoio e incentivo constantes.

## RESUMO

Um dos principais aspectos da otimização de circuitos analógicos é a redução de seu tamanho total, que determina a escolha das opções de encapsulamento. Aumentar a densidade de transistores em circuitos integrados é a melhor maneira de reduzir os custos e aumentar a velocidade e a confiabilidade dos mesmos. Estudos cuidadosos devem ser feitos para ajudar a tomada de tais decisões, baseadas na tecnologia CMOS atual. Este é o caso quando as equações gerais, que representam tais circuitos integrados, são complexas – não posinomiais - e em geral representadas por funções não convexas, impossibilitando o uso da programação convexa para a otimização. Desde o lançamento do primeiro microprocessador, o Intel 4004 em 1971, a corrida pela miniaturização foi acelerada e tornou-se ainda mais importante para a indústria. Na mesma época em que os microprocessadores de 32 bits e tecnologia CMOS estavam sendo produzidos, em meados da década de 1980, outra revolução acontecia no mundo da matemática. Até então, os métodos de otimização eram separados em programação linear (centrados no método *simplex*) e programação não linear. Em 1984, Karmarkar propôs o método chamado de *polynomial-time interior method* para a programação linear, seguido no ano posterior por uma conexão com os métodos de *barreiras* e passou a ser aplicado também na minimização de funções objetivo não lineares sujeitas a restrições não lineares. Nos anos mais recentes, importantes livros textos sobre otimização incluíram capítulos sobre o que veio a ser chamado de Métodos dos Pontos Interiores. Isto foi seguido pelo “Optimization Toolbox” do Matlab, cuja versão mais recente (5ª) inclui algoritmos que usam estes métodos. Neste trabalho, o Método dos Pontos Interiores é aplicado objetivando a minimização de áreas de circuitos analógicos CMOS. O “Toolbox” do Matlab é usado para a obtenção das dimensões otimizadas e, em seguida, são realizadas simulações dos circuitos com as referidas dimensões. Os algoritmos e aspectos teóricos mais importantes são comentados. As referências são incluídas para possibilitar estudos adicionais.

**Palavras-Chave:** CMOS, MOSFET; Integrador de Seevinck; Circuitos Analógicos; Otimização; Restrições; Programação Linear; Funções Não Lineares; Métodos dos Pontos Interiores.

## ABSTRACT

A main aspect of the optimization of analog circuits is the reduction of its overall size, which determines the choice of packaging options. Raising the density of transistors on integrated circuits is the best way to reduce costs and increase their speed and reliability. Careful studies have to be performed to help make such decisions, based on current CMOS technology. This is the case when general equations, representing such integrated circuits, are complex – nonposynomials – and usually represented by nonconvex functions, preventing the use of convex programming for the optimization. Since the inception of the first microprocessor, the Intel 4004 in 1971, the race for miniaturization has been accelerated and became even more important for the industry. By the time that 32-bit microprocessors and CMOS technology were being produced, around the mid-80s, another revolution was taking the stage in the mathematics world. Until then, optimization methods were distinctly divided into linear programming (centered in the *simplex* method) and nonlinear programming. In 1984, Karmarkar proposed a method called *polynomial-time interior method* for linear programming, followed in the next year by a connection with *barrier* methods which came to be also applied in the minimization of nonlinear objective functions subject to nonlinear constraints. In more recent years, important textbooks about optimization have included chapters about what came to be called Interior Point Methods. This was followed by Matlab's Optimization Toolbox, whose last version (v.5) includes algorithms dedicated to such problems. Some examples are exhibited in this work, computing the minimization of areas of CMOS analog circuits. The Matlab Optimization Toolbox is used to obtain the optimized dimensions, followed by simulations of the circuits using these dimensions. The main theoretical aspects and algorithms are commented. References are provided for further studies.

**Keywords:** CMOS, MOSFET; Seevinck Integrator; Analog Circuits; Optimization; Constraints; Nonlinear Programming; Nonlinear Functions; Interior-Point Methods.

## LISTA DE ILUSTRAÇÕES

Figura.3.1 — Divisor de tensão a MOSFET.

Figura 3.2 — Integrador de Seevinck balanceado de domínio logarítmico, para tensão de alimentação mínima.

Figura 3.3 — Respostas em frequência de pequenos sinais do integrador de Seevinck para: Projeto A (linha sólida); Projeto B (linha tracejada); Projeto C (linha pontilhada).

## **LISTA DE TABELAS**

Tabela 3.1.1 — Resultados otimizados para o Divisor de Tensão CMOS

Tabela 3.2.1 — Modelo do MOSFET adotado no projeto

Tabela 3.2.2 — Restrições de espaço de projeto

Tabela 3.2.3 — Resultados otimizados para os exemplos A, B e C

Tabela 3.2.4 — Resultados de simulações para os exemplos A, B e C

## LISTA DE SÍMBOLOS

$f(x)$	função da variável $x$
$C_i$	coeficientes positivos
$a_{ni}$	expoentes reais
$x_i$	variáveis positivas
$h_i(x)$	restrições de igualdades
$g_i(x)$	restrições de desigualdades
$s$	vetor das variáveis de folga
$\bar{X}$	conjunto de pontos factíveis
$x^*$	solução ótima (local ou global)
$\nabla f(x)$	gradiente da função
$\lambda$	multiplicadores de Lagrange para restrições de igualdades (Luenberger)
$\mu$	multiplicadores de Lagrange para restrições de desigualdades (Luenberger)
$S$	matriz diagonal em que $s$ forma os elementos da diagonal
$Z$	matriz diagonal em que $z$ forma os elementos da diagonal
$A_E^T(x)$	matriz Jacobiana das funções de restrições de igualdades
$A_I^T(x)$	matriz Jacobiana das funções de restrições de desigualdades
$y$	multiplicadores de Lagrange para restrições de igualdades (Nocedal)
$z$	multiplicadores de Lagrange para restrições de desigualdades (Nocedal)
$C_E(x)$	vetor das restrições de igualdades
$C_I(x)$	vetor das restrições de desigualdades
$\mathcal{L}$	Lagrangeano
$\nabla_{xx}^2 \mathcal{L}$	Hessiana do Lagrangeano
$p$	passo da iteração
$\alpha_s^{max}$	parâmetro do cálculo do passo
$\alpha_z^{max}$	parâmetro do cálculo do passo
$c^U$	limite superior das funções de restrições
$c^L$	limite inferior das funções de desigualdades
$b^U$	limite superior das variáveis
$b^L$	limite inferior das variáveis

$V_P$	tensão de “pinch-off”
$V_{SB}$	tensão fonte-substrato
$\phi_t$	potencial termodinâmico
$I_D$	corrente de dreno
$I_S$	corrente específica
$V_{GB}$	tensão port-substrato
$V_{T0}$	tensão de limiar no equilíbrio
$n$	fator de rampa
$\mu$	mobilidade dos portadores de carga
$C'_{ox}$	capacitância por unidade de área do óxido
$W$	largura do canal
$L$	comprimento do canal
$f_o$	função objetivo
$V_X$	potencial de saída do divisor de tensão CMOS
$V_{SS}$	tensão de alimentação negativa
$I_{SSq}$	corrente específica do transistor quadrado
$V_{DD}$	tensão de alimentação positiva
$i_f$	corrente de saturação direta normalizada
$a$	parâmetro de ajuste da expressão da corrente de dreno em inversão fraca
$b$	parâmetro de ajuste da expressão da corrente de dreno em inversão fraca
$i_r$	corrente de saturação reversa normalizada
$V_{DB}$	tensão dreno-substrato
$I$	corrente de polarização no integrador de Seevinck
$I_0$	corrente de polarização no integrador de Seevinck
$I_{BIAS}$	corrente de polarização no integrador de Seevinck
$i_{INmáx}$	máximo valor da corrente de entrada
$i_{INmin}$	mínimo valor da corrente de entrada
$C$	capacitância externa do integrador de Seevinck
$C_0$	capacitância metal-óxido-metal por unidade de área
$K_{INT}$	constante de integração
$A_{tran}$	soma das áreas ativas dos transistores no integrador de Seevinck
$A_{cap}$	soma das áreas dos capacitores no integrador de Seevinck

## SUMÁRIO

<b>1. INTRODUÇÃO</b>	14
<b>2. MÉTODOS DOS PONTOS INTERIORES</b>	18
2.1 ORIGENS HISTÓRICAS	18
2.2 CONDIÇÕES BÁSICAS DOS PROBLEMAS NÃO LINEARES	19
2.3 CONDIÇÕES DE OTIMALIDADE PARA RESTRIÇÕES NÃO LINEARES	21
2.4 O ALGORITMO PRIMAL-DUAL BÁSICO	24
2.5 KNITRO – TOOLBOX DO MATLAB PARA PROBLEMAS NÃO LINEARES	26
<b>3. FORMULAÇÃO DOS PROBLEMAS</b>	29
3.1 PROJETO DO DIVISOR DE TENSÃO	29
3.1.1 Problema de Otimização	30
3.1.2 Procedimentos para Otimização	31
3.1.3 Resultados do projeto do divisor de tensão	32
3.1.4 Conclusões sobre o projeto do divisor de tensão	34
3.2 PROJETO DO INTEGRADOR DE SEEVINCK	35
3.2.1 Problema de Otimização	36
3.2.2 Procedimentos para a Otimização	41
3.2.3 Resultados do projeto do Integrador SEEVINCK	42
3.2.4 Conclusões sobre o projeto do Integrador de SEEVINCK	47
<b>4. CONCLUSÕES FINAIS</b>	48
<b>5. REFERÊNCIAS</b>	52

<b>APÊNDICES</b>	56
APÊNDICE A – Cálculos simbólicos do MAPLE	56
A.1 Projeto do divisor de tensão	56
A.2 Projeto do Integrador de Seevinck	63
APÊNDICE B – Rotinas do Toolbox do MATLAB	74
B.1 Projeto do divisor de tensão	74
B.2 Projeto do Integrador de Seevinck	83
APÊNDICE C – Cálculos das iterações	95
C.1 Projeto do divisor de tensões	95
C.2 Projeto do Integrador de Seevinck (Exemplos A, B e C)	120

# 1. INTRODUÇÃO

Um dos principais aspectos da miniaturização de circuitos integrados analógicos é o tamanho total (área de silício), que determina o custo de fabricação e a escolha das opções de encapsulamento. Aumentar a densidade de transistores em circuitos integrados, ou seja, distribuir um número maior de componentes numa área menor, é a melhor maneira de reduzir os custos e aumentar a velocidade e a confiabilidade. Estudos cuidadosos devem ser feitos para ajudar a tomada de tais decisões, baseadas na tecnologia CMOS atual.

Desde a introdução do primeiro microprocessador, o Intel 4004 em 1971, a corrida pela miniaturização foi acelerada e se tornou ainda mais importante para a indústria. Na ocasião em que os microprocessadores em tecnologia CMOS com 32-bits eram produzidos, na metade dos anos 1980, outra revolução estava surgindo no mundo da Matemática. Anteriormente, os métodos de otimização com restrições eram divididos distintamente em programação linear (centrada no método simplex) e em programação não linear. Em 1984, Karmarkar (KARMAKAR, 1984, p. 373-395) anunciou um método para programação linear que chamou de “polynomial-time interior method”, que foi seguido, no ano posterior, por uma conexão formal entre este método e os métodos das barreiras.

As últimas duas décadas trouxeram uma aceleração da pesquisa em programação (otimização) não linear tratando a minimização de funções objetivo (lineares ou não) sujeitas a restrições não lineares, presenças constantes nos circuitos analógicos (BYRDa, 1998, Passim) (BYRDb, 1998, Passim) (BYRDc, 1998, Passim) (CURTIS, 2010, Passim).

Atualmente, os pesquisadores consideram a programação linear como um caso especial da programação não linear. Nos anos mais recentes, importantes livros textos sobre otimização incluíram capítulos sobre o que veio a ser chamado de Métodos dos Pontos Interiores. Isto foi seguido pelo “Optimization Toolbox” do Matlab, cuja versão mais recente (5<sup>a</sup>) inclui algoritmos que usam estes métodos. Estas rotinas facilitam bastante a tarefa de achar valores ótimos para o circuito, os

quais, para confirmação do atendimento das especificações ou refinamento do projeto, podem ser aplicados, posteriormente, em programas de simulação de circuitos eletrônicos, por meio dos quais se pode proceder a simulações DC, AC, no domínio do tempo, do ponto de operação, de ruído, entre outras, muitas das quais com saída gráfica. Isto é especialmente importante nos casos que envolvem funções não convexas e não lineares, nos quais os valores iniciais escolhidos são, por vezes, cruciais para a pesquisa de mínimos locais adequados.

O uso de um modelo genérico para todos os níveis de inversão do MOSFET, ao invés do modelo quadrático que somente é válido em inversão forte, aumenta o campo de trabalho para que se ache uma solução ótima e evita soluções incorretas advindas do uso de modelos de inversão forte quando o circuito opera otimamente em regiões de inversão moderada ou fraca. Por outro lado, as equações gerais são mais complexas – não posinomiais – e as restrições são representadas por funções não convexas, impossibilitando o uso da programação convexa para a otimização (HERSHENSON, 2001, Passim) (MANDAL, 2001, Passim) (OLIVEIROS, 2008, Passim) (VANDERHAEGEN, 2004, Passim).

Antes de prosseguir na questão dos pontos interiores, é mister uma rápida revisão de alguns conceitos básicos, inicialmente pelo esclarecimento que, em nossas análises, são consideradas apenas as minimizações, isto é, a busca pelos pontos em que a função objetivo tem os menores valores. Estes pontos são chamados de mínimos locais e definidos matematicamente por derivadas da função objetivo iguais a zero naqueles pontos. Um ponto ótimo, neste caso, seria o ponto mínimo de menor valor, podendo ser considerado como mínimo global entre determinados limites de valores das variáveis da função objetivo. Porém, deve-se levar em conta que nem sempre se consegue achar todos os pontos mínimos.

Os métodos para otimização de funções não lineares precisam ser avaliados cuidadosamente. Por exemplo, o método chamado de Programação Geométrica (SHARMA, 2006, p. 49-53) precisa que a função objetivo não linear seja *posinomial*, definida como:

$$f(x) = \sum_{i=1}^N C_i \cdot x_1^{a_{1i}} \cdot x_2^{a_{2i}} \dots x_n^{a_{ni}} \quad (1.1)$$

onde,  $C_i$  são coeficientes positivos,

$a_{1i}, a_{2i}, \dots, a_{ni}$  são expoentes reais,

$x_1, x_2, \dots, x_n$  são variáveis positivas,

$N$  é o número de variáveis.

Assim, a Programação Geométrica só deve ser aplicada efetivamente se o problema puder ser convertido numa função posinomial, que tenha coeficientes positivos em cada componente e variáveis positivas cujos expoentes sejam reais.

No caso da Programação Convexa (BOYD, 2004, p. 136-137), esta é limitada aos problemas de otimização em que as funções objetivo e de desigualdades são convexas (o que pode ser analisado pelo cálculo de suas Hessianas), além das funções de igualdades, que devem ser *afins* (“*affine*”).

Neste trabalho, os métodos dos pontos interiores são adotados para achar soluções otimizadas para o projeto de circuitos analógicos com a tecnologia CMOS, desde que este projeto seja caracterizado por problemas de otimização não lineares e não convexas.

Alguns exemplos da aplicação destes métodos são então apresentados, nos quais se busca a minimização de áreas de circuitos analógicos visando implementação na tecnologia IBM 0.13. Nas simulações subsequentes é utilizado o modelo BSIM3v3.1, com os parâmetros desta tecnologia.

Os seguintes projetos são considerados:

- Projeto de divisor de tensão a MOSFET, utilizando as equações do modelo ACM (“Advanced Compact MOSFET model”) (MANIERO, 2003, Passim).
- Projeto do Integrador “Log-Domain” para mínima tensão de alimentação, em tecnologia CMOS, de Seevinck (SEEVINCK, 2000, Passim) (SILVA, 2011, Passim) (SOBRAL, 2008, Passim).

O “Toolbox” de otimização do MATLAB (THE MATHWORKS, 2010, Passim) é usado para achar as dimensões ótimas em cada problema e simulações dos circuitos são realizadas com estes resultados otimizados por meio do simulador SMASH (DOLPHIN, 2008, Passim).

## 2. MÉTODOS DOS PONTOS INTERIORES

### 2.1 Origens históricas

A busca de soluções para problemas de otimização não lineares tem suas origens históricas nos séculos XVIII e XIX, com os trabalhos de Euler e Lagrange, ao estudarem o *cálculo das variações* (GRIVA, 2009, p. 543). Neste cálculo os parâmetros são funções ao invés de variáveis. Até esta época não se conhecia como calcular derivadas de funções e Euler só obteve resultados nos anos 40 do século XVIII (GRIVA, 2009, p. 544).

Na mesma época Lagrange sugeriu que as soluções passassem por uma “perturbação” de  $x(t)$  para  $x(t) + \epsilon y(t)$ , onde  $\epsilon$  é um número pequeno e  $y(t)$  é uma função arbitrária que satisfaz  $y(t_1) = y(t_2) = 0$ , tendo provado que esta função perturbada ainda representava um passo (“path” ou “step”) entre dois pontos. Isto lhe possibilitou a análise dos problemas usando o cálculo usual e suas derivadas, considerando apenas as restrições de igualdades.

Daí surgiu o conceito de “multiplicadores” (escalares ou funções de uma variável independente  $t$ ), hoje conhecidos como *multiplicadores de Lagrange*, apesar de terem sido propostos por Euler (GRIVA, 2009, p. 545).

Os próximos passos significativos desta pesquisa só ocorreram muito mais tarde, quando Karush, em 1939, e Kuhn e Tucker em 1951 divulgaram suas pesquisas, envolvendo restrições de desigualdades e definindo as condições de otimalidade dos problemas de dimensões finitas, que passaram a ser conhecidas como *condições de Karush-Kuhn-Tucker (K-K-T)* (GRIVA, 2009, p. 545). Este trabalho foi antecedido, na década de 1870, pela introdução da variável folga (“*slack*”) com que Weierstrass (Apud GRIVA, 2009, p. 546) convertia uma restrição de desigualdade  $g(x) \leq 0$  numa restrição de igualdade equivalente  $g(x) + s^2 = 0$ , em que a variável folga  $s$  era elevada ao quadrado, evitando valores negativos (WRIGHT, 2004, Passim).

Novo intervalo ocorreu, até meados da década de 1980, como mencionado, para que ocorressem novos avanços importantes no estudo da otimização de problemas não lineares.

## 2.2 Condições básicas dos problemas não lineares

Nossa área de estudos concentra-se na otimização de problemas não lineares e não convexos com restrições de igualdades e desigualdades. Sua forma geral é:

$$\begin{aligned} & \underset{x}{\text{Minimizar}} && f(x) \\ & \text{Sujeita a} && h_i(x) = 0, \quad i=1, \dots, m \\ & && g_i(x) \leq 0, \quad i=1, \dots, n \end{aligned} \quad (2.2.1)$$

onde:  $x \in \mathbb{R}^{m+n}$ ,  $h_i(x) \in \mathbb{R}^m$ ,  $g_i(x) \in \mathbb{R}^n$  (LUENBERGER, 2008, p. 469).

Para eliminar as restrições de desigualdades, esta forma pode ser transformada na forma:

$$\begin{aligned} & \underset{x,s}{\text{Minimizar}} && f(x) \\ & \text{Sujeita a} && h_i(x) = 0, \quad i=1, \dots, m \\ & && g_i(x) - s_i = 0, \quad i=1, \dots, n \\ & && s_i \geq 0 \end{aligned} \quad (2.2.2)$$

em que, através do vetor das variáveis de folga  $s$ , as restrições de desigualdades foram transformadas em igualdades.

Como nosso campo de estudos é a minimização de áreas de circuitos, o campo de buscas será o conjunto de *pontos factíveis*  $\bar{X}$ , isto é, aqueles que atendem a todas as restrições, denominado de *região factível*. Portanto, a solução global  $x^*$  de  $f(x)$  é a que satisfaz  $f(x^*) \leq f(x)$  para todos os  $x \in \bar{X}$ . O caso mais viável em nosso campo de estudos é a solução em termos de *mínimos locais*, definidos como:  $x^*$  é um mínimo local de  $f(x)$  em  $\bar{X}$  desde que satisfaça  $f(x^*) \leq f(x)$  para todos os  $x \in \bar{X}$  tal que  $\|x - x^*\| < \varepsilon$ , sendo  $\varepsilon$  o erro considerado.

Assume-se que a função objetivo  $f(x)$  e as funções de restrições  $g_i$  e  $h_i$  são dupla e continuamente diferenciáveis. Como tratamos de problemas não lineares e não convexos, para isto ocorrer basta que ou a função objetivo ou uma das funções de restrições seja não linear. Similarmente, basta que uma destas funções seja não convexa para que o problema seja considerado não convexo.

Considera-se que a solução  $x^*$  de um problema de otimização é um ponto *regular*, isto é, os gradientes das restrições neste ponto são linearmente independentes.

Assume-se também que:

- os gradientes das funções de igualdades  $\{\nabla h_i(x), i=1, \dots, m\}$  são linearmente independentes;
- os gradientes das funções de desigualdades *ativas* em  $X^*$ ,  $\{\nabla g_i(\bar{x}) : g_i(\bar{x}) = 0\}$  são linearmente independentes.

Um ponto factível  $\bar{x}$  é dito no *limite* (“*boundary*”) da restrição de desigualdade quando  $g_i(\bar{x}) = 0$ . Se  $g_i(\bar{x}) > 0$  (a restrição de desigualdade é inativa), este ponto é dito no *interior* da restrição. As restrições de igualdades são consideradas ativas em qualquer ponto factível. O *conjunto ativo* (“*active set*”) em um ponto factível é definido como o conjunto de todas as restrições ativas naquele ponto.

Um conjunto de pontos factíveis para os quais exista pelo menos uma restrição de desigualdade ativa comum é chamado de *limite da região factível*. Todos os outros pontos factíveis são denominados de *pontos interiores* com relação às desigualdades; todos os pontos factíveis satisfazem as restrições de igualdades.

Dadas estas condições, segue-se uma metodologia de busca de pontos factíveis, de maneira que haja uma convergência que garanta a descoberta o mais rapidamente possível de um (ou mais) pontos ótimos locais.

No caso de problemas não lineares, a obtenção de pontos factíveis requer a solução de um sistema de equações e desigualdades não lineares, problema certamente não trivial. Pode ser necessário, por exemplo, usar-se métodos variados ou testar-se determinado método repetidamente com valores iniciais modificados para obter-se um ponto factível.

### 2.3 Condições de otimalidade para restrições não lineares

As condições de otimalidade de primeira ordem, ou condições de K-K-T, de um problema não linear como em (2.2.1), podem ser escritas (LUENBERGER, 2008, p. 469-470) na forma:

$$\begin{aligned}\nabla f(x) + \lambda^T \nabla h(x) + \mu^T \nabla g(x) &= 0, \\ h(x) &= 0, \\ g(x) &\leq 0, \\ \mu^T g(x) &= 0,\end{aligned}\tag{2.3.1}$$

sendo que o vetor  $\mu$  (nas restrições de desigualdades) e o vetor  $\lambda$  (nas restrições de igualdades) são os multiplicadores de Lagrange. A última condição é a de *folga* (“slackness”). Busca-se então a solução para  $(x, \lambda)$ , sendo que  $\lambda$  e  $h$  têm a mesma dimensão.

As condições de otimalidade, adotando nomenclatura um pouco diferente e considerando as variáveis de folga, são (NOCEDAL, 2006, p. 564-565):

$$\begin{aligned}\nabla f(x) - A_E^T(x)y - A_I^T(x)z &= 0, \\ SZ - \mu e &= 0, \\ c_E(x) &= 0, \\ c_I(x) - s &= 0,\end{aligned}\tag{2.3.2}$$

sendo  $S$  e  $Z$  matrizes diagonais cujos elementos das diagonais são  $s$  e  $z$  respectivamente;  $c_E(x)$  e  $c_I(x)$  representando as restrições de igualdades e desigualdades, respectivamente.

Temos:  $e = (1, 1, \dots, 1)^T$ , com  $\mu > 0$ ,  $s \geq 0$  e  $z \geq 0$ .

Definimos:

- $A_E^T(x)$  e  $A_I^T(x)$  como sendo as matrizes Jacobianas das funções  $c_E(x)$  e  $c_I(x)$ , respectivamente;
- $y$  e  $z$  são seus respectivos multiplicadores de Lagrange, associados às funções;

Esta solução, chamada de “*continuação*” ou “*homotopia*”, utiliza as condições de K-K-T - “*perturbadas*” com as variáveis de folga - e busca resultados aproximados com uma sequência de parâmetros positivos  $\{\mu_k\}$  que convergem para zero, mantendo  $s$  e  $z > 0$ .

Para uma solução  $(x^*, s^*, y^*, z^*)$  e para valores positivos e suficientemente pequenos de  $\mu$ , o problema não linear tem uma solução local única, denotada por  $(x(\mu), s(\mu), y(\mu), z(\mu))$ . A trajetória percorrida por estes pontos é chamada de *caminho* (“*path*”) *central primal-dual e converge para*  $(x^*, s^*, y^*, z^*)$  na medida em que  $\mu \rightarrow 0$ . (NOCEDAL, 2006, cap. 12).

Para o caso da solução por *barreira*, associamos (2.2.2) com o problema de barreira (NOCEDAL, 2006, p. 565-566):

$$\begin{aligned} \underset{x, s}{\text{Minimizar}} \quad & f(x) - \mu \sum_{i=1}^m \log s_i \\ \text{Sujeita a} \quad & c_E(x) = 0, \\ & c_I(x) - s = 0, \end{aligned} \tag{2.3.3}$$

onde  $\mu > 0$  e  $\log$  é o logaritmo natural.

As condições de K-K-T são:

$$\begin{aligned} \nabla f(x) - A_E^T(x)y - A_I^T(x)z &= 0, \\ -\mu S^{-1}e + z &= 0, \\ c_E(x) &= 0, \end{aligned} \tag{2.3.4}$$

$$c_i(x) - s = 0,$$

em que a segunda equação, cujos elementos diagonais da matriz  $S$  são positivos, ao ser multiplicada por  $S$ , torna as condições K-K-T da solução por barreira coincidentes com as da solução por homotopia.

Desta forma, as soluções por homotopia definem as direções do primal-dual e as por barreira conduzem às iterações globalmente convergentes.

As diferentes formas de abordagem destes problemas de pontos interiores geram algoritmos diversos e, em certos casos, específicos. Atualmente, o que se considera são classes de métodos de pontos interiores.

**2.4 O Algoritmo Primal-Dual básico** (BALBO, 2010, Passim) (FORSGREN, 2002, Passim) (GERTZ, 2004, Passim) (GOULD, 2005, Passim) (GRIVA, 2009, p. 641-647) (GUARDIA, 2008, Passim) (VANDERBEI, 1991, Passim).

Tomando por base as condições K-K-T da solução da homotopia e aplicando o método de Newton com as variáveis  $x$ ,  $s$ ,  $y$  e  $z$ , obtemos o *sistema Primal-Dual* (NOCEDAL, 2006, p. 566-568), também conhecido por “*direct step*” ou “*Newton step*”:

$$\begin{bmatrix} \nabla_{xx}^2 \mathcal{L} & 0 & -A_E^T(x) & -A_I^T(x) \\ 0 & Z & 0 & S \\ A_E(x) & 0 & 0 & 0 \\ A_I(x) & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_s \\ p_y \\ p_z \end{bmatrix} = - \begin{bmatrix} \nabla f(x) - A_E^T(x)y - A_I^T(x)z \\ Sz - \mu e \\ c_E(x) \\ c_I(x) - s \end{bmatrix}, \quad (2.4.1)$$

onde  $\mathcal{L}(x, s, y, z) = f(x) - y^T c_E(x) - z^T (c_I(x) - s)$  é o *Lagrangiano* do programa não linear. O vetor  $p = [p_x, p_s, p_y, p_z]$  é o *passo* que, ao ser calculado, determina a nova iteração  $(x^+, s^+, y^+, z^+)$ .

As iterações são realizadas considerando-se os passos  $p$  e parâmetros escolhidos de acordo com o método a ser seguido para o problema em pauta.

Temos para cada nova iteração:

$$x^+ = x + \alpha_s^{max} p_x, \quad (2.4.2a)$$

$$s^+ = s + \alpha_s^{max} p_s, \quad (2.4.2b)$$

$$y^+ = y + \alpha_z^{max} p_y \quad (2.4.2c)$$

$$z^+ = z + \alpha_z^{max} p_z, \quad (2.4.2d)$$

onde,  $\alpha_s^{max} = \max\{\alpha \in (0, 1] : s + \alpha p_s \geq (1 - \tau)s\}$ ,

$$\alpha_z^{max} = \max\{\alpha \in (0, 1] : z + \alpha p_z \geq (1 - \tau)z\},$$

sendo,  $\tau \in (0, 1)$ , (podendo ser adotado o valor de 0,995).

Os valores calculados de  $\alpha$  evitam que as variáveis  $s$  e  $z$  cheguem aos limites inferiores (zero) muito rapidamente.

Esta iteração é a base dos métodos dos pontos interiores modernos. Quando o algoritmo não consegue fazer esta iteração, por exemplo, quando não há convexidades locais nas proximidades, o passo da iteração passa a ser calculado por Gradientes Conjugados numa região confiável (“*trust region*”) e  $x$  e  $s$  são ajustados, mantendo os valores de  $s$  positivos.

A cada iteração, o algoritmo deve decrescer uma função de mérito (“*merit function*”) tal como:

$$f_{\mu}(x, s) + v \llbracket (c_E(x), c_I(x) + s) \rrbracket, \quad (2.4.3)$$

em que o parâmetro  $v$  aumenta a cada iteração, forçando a solução para a região factível. Se a função de mérito não diminui, o passo atual deve ser rejeitado e um novo deve ser calculado.

A escolha dos parâmetros de barreira  $\{\mu_k\}$  também constitui um procedimento importante nos algoritmos. Eles podem permanecer constantes durante uma fase das iterações, enquanto as condições K-K-T não forem atendidas e variarem após este atendimento.

O processo de otimização precisa levar em conta as situações inerentes aos problemas abordados, especialmente as não linearidades e singularidades. A literatura já existente descreve este processo para várias situações.

Este problema é resolvido, em sua totalidade, através do Toolbox KNITRO do MATLAB (THE MATHWORKS, 2010, Passim) (WALTZ, 2010, Passim). Como consequência da diversidade de abordagens dos métodos de pontos interiores, as abordagens das rotinas programadas em computador também são diversas. O KNITRO adota, em princípio, o que foi descrito acima, como se verá nos exemplos realizados.

## 2.5 KNITRO — Toolbox do MATLAB para problemas não lineares

Conforme descrito no manual do usuário na introdução de sua versão 7.0 de Setembro de 2010 (WALTZ, 2010, Passim), o “software” KNITRO, parte do Toolbox de Otimização do MATLAB em sua versão 5.0, em ambiente Windows/XP. Seu objetivo principal é “achar soluções locais para problemas contínuos, não lineares e de grande porte.” Também é eficaz na solução dos métodos de pontos interiores (“barreira”).

Os problemas para o KNITRO têm a forma:

$$\begin{aligned} & \text{Minimizar } f(x) \\ & \quad x \\ & \text{Sujeita a } c^L \leq c(x) \leq c^U \\ & \quad b^L \leq x \leq b^U \end{aligned} \tag{2.5.1}$$

onde  $x \in \mathbb{R}^n$  engloba as variáveis,  $c^L$  e  $c^U$  são limites (caso hajam) inferiores e superiores das restrições;  $b^L$  e  $b^U$  são limites inferiores e superiores (caso hajam) das variáveis. Os problemas com descontinuidades nas derivadas podem ser aceitos.

O algoritmo que será usado é chamado de “*interior-point*” (“Interior/Direct algorithm”), em que o KNITRO divide o problema não linear em vários problemas de barreira controlados pelo parâmetro de barreira  $\mu$ . A convergência é estimulada por *regiões confiáveis* (“trust regions”) e *funções de mérito* (“merit functions”). As iterações são repetidas até que a função objetivo atinja uma tolerância desejada.

O KNITRO deixa ao usuário a opção de calcular os valores exatos da derivada da função objetivo, do Jacobiano (matriz das derivadas parciais de primeira ordem das restrições) e da Hessiana da função Lagrangeana. A informação destes valores ao programa permite maior eficiência, pois o programa os calcula por aproximações de diferenças finitas.

Há uma gama de opções que permitem ao usuário modificar o comportamento do programa. Entre as opções, citamos algumas das mais importantes:

- tolerância para a factibilidade do modelo (valor “*default*”: 1.0e-8);
- tolerância para o erro da otimalidade ou condições de K-K-T (valor “*default*”: 1.0e-6);
- tolerância para a mudança relativa entre os valores calculados para o ponto ótimo  $x^*$  (valor “*default*”: 1.0e-15);
- opção para que o programa calcule os gradientes da função objetivo e das restrições (“*forward finite-differences*” ou “*centered finite-differences*”);
- opção para que o programa calcule a Hessiana da função Lagrangeana (“*dense quasi-Newton bfgs*”: usa informações dos gradientes para calcular uma aproximação da matriz Hessiana simétrica positiva-definida; “*dense quase-Newton SR1*”: semelhante à anterior, porém não necessariamente positiva-definida);

Como o KNITRO acha um ponto mínimo local (se houver) em cada ocasião em que é usado com um determinado conjunto de dados iniciais (variáveis, constantes, limites, tolerâncias, etc.), há, naturalmente, uma forte correspondência entre estes valores iniciais e o ponto mínimo local resultante das iterações. O usuário, a depender do problema proposto (quando, por exemplo, o ponto achado é infactível localmente), pode ter de lançar mão de alternativas para estes valores iniciais – especialmente as variáveis - na tentativa de encontrar outros pontos mínimos locais. Dependendo do problema, a probabilidade de se achar um ponto mínimo local melhor aumenta com a quantidade de valores iniciais testados. Neste trabalho, o ponto mínimo local encontrado que fornece a função objetivo de menor valor é considerado como um mínimo global aceitável.

Os exemplos a serem apresentados no capítulo 3 mostrarão estas questões utilizando projetos de circuitos analógicos e suas necessidades de minimização das respectivas áreas. Os cálculos são feitos inicialmente usando o programa MAPLE (WATERLOO MAPLE, 2010, Passim), para os cálculos dos gradientes e Hessianas, propiciando uma melhor leitura das funções, vetores e matrizes. Estes dados são passados para o MATLAB/KNITRO para as devidas iterações e resultados. Estes cálculos estão reproduzidos nos apêndices, conforme seus formatos originais.

Simulações gráficas são realizadas pelo programa Smash (DOLPHIN INTEGRATION, 2008, Passim), que propicia uma visão global do problema, seus parâmetros e soluções.

## 3. FORMULAÇÃO DOS PROBLEMAS

### 3.1 Projeto do divisor de tensão

Um divisor de tensão a MOSFET é um circuito extremamente simples que serve para a obtenção de tensões de referência a serem aplicadas em nós de alta impedância, como por exemplo, terminais de porta em fontes ou sorvedores de corrente. Nesta abordagem, vamos nos limitar a um divisor de tensão formado por apenas duas cargas ativas, um transistor MOS canal P e um transistor canal N, em conexão diodo (terminais de porta e dreno conectados), como ilustrado na Fig.3.1.

De acordo com o modelo ACM (“Advanced Compact MOSFET model”) (CUNHAa, 1998, Passim), desprezados efeitos de canal curto, a corrente de dreno  $I_D$  no regime de saturação é função das tensões terminais através da expressão:

$$V_P - V_{SB} = (+/-)\phi_t \left[ \sqrt{1 + \frac{I_D}{I_S}} - 2 + \ln \left( \sqrt{1 + \frac{I_D}{I_S}} - 1 \right) \right] \quad (3.1.a)$$

onde  $V_P$  é a tensão de “pinch-off”, dada aproximadamente por:

$$V_P = \frac{V_{GB} - V_{T0}}{n} \quad (3.1.b)$$

e  $I_S$  é a corrente específica, dada por:

$$I_S = \mu C'_{ox} \phi_t^2 \frac{n W}{2 L} \quad (3.1.c)$$

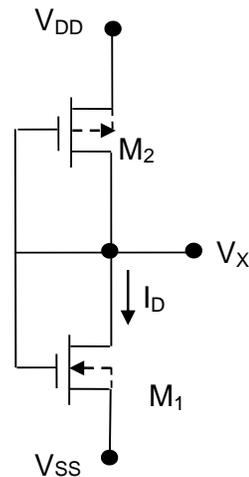
Nas expressões (3.1),  $V_{SB}$  e  $V_{GB}$  são as tensões fonte-substrato e porta-substrato, respectivamente,  $V_{T0}$  é a tensão de limiar no equilíbrio,  $n$  é o fator de rampa (aqui aproximado por um valor constante, entre 1 e 2),  $\mu$  é a mobilidade dos portadores de carga,  $C'_{ox}$  é a capacitância por unidade de área do óxido,  $W$  é a largura do canal,  $L$  é o comprimento do canal e  $\phi_t$  é o potencial termodinâmico. A expressão é válida para todo o regime de inversão (fraca, moderada e forte). O sinal entre parênteses do lado direito da igualdade é + (mais) para transistores canal N e – (menos) para transistores canal P.

De acordo com as expressões (3.1), escolhida uma corrente de dreno  $I_D$ , comum aos dois transistores, suas razões de aspecto ( $W/L$ ) podem ser calculadas

numericamente para obter a tensão porta-substrato de cada um deles, por sua vez, relacionada ao potencial no nó de saída do divisor de tensão.

Entretanto, o dimensionamento deste circuito sem uma técnica de otimização pode levar a um mau aproveitamento da área de silício. No item seguinte, o problema de otimização relativo a este circuito é formulado, visando a minimização da área.

Figura.3.1 — Divisor de tensão a MOSFET



### 3.1.1 Problema de Otimização

Para dimensionar os transistores do divisor de tensão CMOS da Fig.3.1, definimos como variáveis de projeto:  $I_D$ ,  $W_1$ ,  $L_1$ ,  $W_2$ ,  $L_2$ . A função objetivo consiste da área ativa:

$$f_O = W_1 L_1 + W_2 L_2 \quad (3.1.1a)$$

e deve ser minimizada, submetida às restrições de igualdades:

$$V_X = V_{SS} + V_{T0N} + n_N \phi_t \left[ \sqrt{1 + \frac{I_D}{I_{SsqN} \frac{W_1}{L_1}}} - 2 + \ln \left( \sqrt{1 + \frac{I_D}{I_{SsqN} \frac{W_1}{L_1}}} - 1 \right) \right] \quad (3.1.1b)$$

e

$$V_X = V_{DD} + V_{T0P} - n_P \phi_t \left[ \sqrt{1 + \frac{I_D}{I_{SsqP} \frac{W_2}{L_2}}} - 2 + \ln \left( \sqrt{1 + \frac{I_D}{I_{SsqP} \frac{W_2}{L_2}}} - 1 \right) \right] \quad (3.1.1c)$$

onde o potencial  $V_X$  é a saída desejada (especificada) do divisor de tensão CMOS ilustrado na Fig.3.1, o índice N(P) denota parâmetro do transistor canal N(P) e  $I_{Ssq}$  é a corrente específica de um transistor quadrado ( $W = L$ ).

Em conformidade com a tecnologia IBM 0.13, cujo mínimo comprimento de canal permitido é  $0,13 \mu\text{m}$  e cuja largura mínima de canal é  $0,16 \mu\text{m}$ , utilizamos os seguintes valores de parâmetros e de tensões de alimentação:

$$\begin{aligned} V_{T0N} &= 0,362 \text{ V}, V_{T0P} = -0,3554 \text{ V}, \\ I_{SsqN} &= 353,38 \text{ nA}, I_{SsqP} = 49,514 \text{ nA}, \\ n_N &= 1,3790, n_P = 1,2952 \text{ e} \\ V_{DD} &= -V_{SS} = 0,6 \text{ V}. \end{aligned}$$

Os parâmetros  $V_{T0N(P)}$ ,  $I_{SsqN(P)}$  e  $n_{N(P)}$  foram extraídos a partir de características corrente-tensão simuladas, utilizando o modelo BSIM3v3.1, pela aplicação da metodologia descrita em (CUNHAb, 2005, Passim).

### 3.1.2 Procedimentos para a Otimização

A otimização foi realizada usando o "Toolbox" de Otimização do MATLAB [THE MATHWORKS, 2010, Passim], aí incluída uma versão de avaliação do pacote Knitro (WALTZ, 2010, Passim), que usa métodos de Pontos Interiores. A ferramenta de otimização foi aplicada com as seguintes especificações, conforme a rotina do MATLAB:

Valor de "Vx" (em Volts): a ser inserido pelo usuário da ferramenta.

Quantidade máxima de iterações do KNITRO: 50.

Valores "default" das tolerâncias (MATLAB), entre os valores de cada iteração:

TolX=  $1\text{e-}15$  (variáveis);

TolFun= $1\text{e-}06$  (função objetivo);

TolCon= $1\text{e-}06$  (região factível das restrições);

Limites ("bounds") inferiores e superiores:

$2.700 \times 10^{-7} \leq L_1 \leq 1.000 \times 10^{-4}$  (metros)

$$2.700 \times 10^{-7} \leq L_2 \leq 1.000 \times 10^{-4} \text{ (metros)}$$

$$2.700 \times 10^{-7} \leq W_1 \leq 1.000 \times 10^{-4} \text{ (metros)}$$

$$2.700 \times 10^{-7} \leq W_2 \leq 1.000 \times 10^{-4} \text{ (metros)}$$

$$1.000 \times 10^{-12} \leq I_D \leq 1.000 \times 10^{-4} \text{ (Amperes)}$$

O limite inferior escolhido para o comprimento (largura) do canal de cada transistor é um valor um pouco maior que o dobro do comprimento (largura) mínimo especificado para a tecnologia. Esta medida pretende evitar que os efeitos de canal curto fiquem muito pronunciados.

As iterações foram calculadas, adotando-se para valores iniciais as 32 combinações entre os valores máximos e mínimos de cada uma das cinco variáveis  $L_1$ ,  $L_2$ ,  $W_1$ ,  $W_2$  e  $I_D$ , conforme a rotina do MATLAB transcrita no Apêndice B.1, a fim de analisar a variabilidade das soluções (mínimos locais da função objetivo), e escolher como solução definitiva a mínima entre todas encontradas.

.

### 3.1.3 Resultados do projeto do divisor de tensão

Mediante a aplicação do Método dos Pontos Interiores, por meio do “Toolbox” de Otimização do MATLAB, ao problema do divisor de tensão CMOS, com os dispositivos representados pelo modelo ACM, observou-se que os resultados obtidos para a função objetivo foram idênticos para todas as combinações dos valores iniciais. Este fato revela a robustez do método para o caso específico deste problema, em que, apesar de não ser possível aplicar um método de programação convexa, a função objetivo parece apresentar um comportamento convexo.

Na Tabela 3.1.1 são apresentadas os resultados otimizados, obtidos pela aplicação da 32ª combinação de valores iniciais, para o caso de 4 projetos, onde os valores especificados são:  $V_X = 0,35 \text{ V}$ ,  $V_X = 0,25 \text{ V}$ ,  $V_X = 0$ ,  $V_X = -0,2 \text{ V}$ . Nas quatro primeiras linhas são apresentadas os valores das larguras e comprimentos de canal dos dois transistores. Nas quinta e sexta linhas são exibidos os valores ótimos da função objetivo (área ativa) e o número de iterações requerido, respectivamente.

Nas sétima e oitava linhas são apresentados os valores da corrente de dreno previstos pela ferramenta de otimização e obtidos por simulação, respectivamente.

Finalmente, na última linha, exibem-se os valores de  $V_x$  alcançados por simulação do ponto de operação. Os resultados de simulação foram extraídos do simulador SMASH [DOLPHIN INTEGRATION INC., 2008, Passim], após terem sido carregados os valores das dimensões ótimas dos transistores.

Os erros calculados nas iterações das otimizações em todos os exemplos, através da verificação dos valores das igualdades em (3.1.1b) e (3.1.1c), estão dentro das tolerâncias especificadas (isto é, as “default” do MATLAB).

A título de ilustração, uma parte dos resultados, com as iterações das combinações de 1 a 3 e de 30 a 32, está transcrita no Apêndice C.1, para o caso do exemplo A (especificação de  $V_x$  igual a 0,35 volts).

**Tabela 3.1.1 — Resultados Otimizados para o Divisor de Tensão CMOS**

<b>Variáveis</b>	<b>Exemplo A <math>V_x = 0,35 \text{ V}</math></b>	<b>Exemplo B <math>V_x = 0,25 \text{ V}</math></b>	<b>Exemplo C <math>V_x = 0 \text{ V}</math></b>	<b>Exemplo D <math>V_x = -0,2 \text{ V}</math></b>
$L_1$ ( $\mu\text{m}$ )	0,43	1,52	11,75	36,05
$L_2$ ( $\mu\text{m}$ )	83,82	70,51	46,53	16,10
$W_1$ ( $\mu\text{m}$ )	94,98	86,32	69,69	55,21
$W_2$ ( $\mu\text{m}$ )	5,26	14,48	44,38	66,94
Área ( $\mu\text{m}^2$ )	482,51	1.151,69	2.884,16	3.067,89
número de iterações	16	11	18	19
$I_D$ ( $\mu\text{A}$ )	6,134	14,24	19,83	14,08
$I_D$ ( $\mu\text{A}$ ) (simulada)	8,313	25,34	48,61	31,27
$V_x$ (V) (simulada)	0,333 V	0,279 V	-0,036 V	-0,318 V

### 3.1.4 Conclusões sobre o projeto do divisor de tensão

A aplicação do método dos pontos interiores para a minimização da área de um divisor de tensão CMOS mostrou-se uma vantajosa alternativa aos métodos de programação convexa, uma vez que permitem a adoção de um modelo genérico para a corrente de dreno do transistor MOS, válido em todo o regime de inversão (fraca, moderada e forte). O método de programação convexa requer a verificação da convexidade não apenas da função objetivo, mas também das restrições pertinentes ao problema de otimização, o que é em geral admissível quando se utiliza um modelo quadrático para a corrente de dreno. Contudo, este modelo é válido apenas para inversão forte, restringindo o espaço de trabalho ou levando a uma solução possivelmente não ótima.

O nível de inversão de um transistor MOS pode ser calculado como a razão entre a corrente de dreno  $I_D$  e a corrente específica,  $I_S = I_{Ssq} \cdot W/L$ . Considera-se que o MOSFET encontra-se em inversão forte para níveis de inversão superiores a 100, e em inversão fraca para níveis de inversão inferiores a 1. Nos exemplos testados, os níveis de inversão dos transistores  $M_1$  (canal N) são: 0,1065 (A), 1,2627 (B), 23,1927 (C), 21,2827 (D).

Para os transistores  $M_2$  (canal P), os níveis de inversão resultam: 2675,4 (A), 2492,1 (B), 1029,3 (C), 151,894 (D).

Percebe-se, portanto, que a solução ótima conduziu a transistores canal P operando em inversão forte e a transistores canal N operando em inversão fraca ou moderada. Para os transistores canal N, o modelo quadrático não seria adequado pois não leva em consideração a inversão fraca, levando a uma previsão incorreta da melhor solução.

Os resultados de simulação apresentados para o potencial  $V_X$  apresentam pequenos desvios que se devem ao fato de não termos considerado efeitos de segunda ordem, como efeitos de canal curto e de degradação da mobilidade com o campo elétrico transversal no modelo utilizado na rotina de otimização. O simulador, por outro lado, leva em consideração estas não idealidades, cuja modelagem em termos analíticos é complexa.

### 3.2 Projeto do Integrador de Seevinck

O integrador de Seevinck (SEEVINCK, 2000, Passim) (MANIERO, 2003, Passim) (MANIERO, 2004, Passim) (SOBRAL, 2008, Passim), de domínio logarítmico e de tensão de alimentação mínima, contém dispositivos que operam em inversão fraca, de modo que as equações do modelo, as condições da polarização (“bias”) requerida e a arquitetura do circuito apontam para equações gerais não posinomiais e restrições que são representadas por funções não convexas. Esta é uma importante célula do circuito para ser usada em filtros analógicos com baixa potência, baixa tensão, em modo corrente, sendo de grande interesse para a utilização de implantes de dispositivos biomédicos (MANIERO, 2004, Passim). Uma vez que os transistores MOSFET devem operar em inversão fraca para a operação correta do integrador de Seevinck, as características I-V são exponenciais. Portanto, as equações do modelo e as restrições do desempenho constituem um problema que é necessariamente não convexo.

O modelo de inversão fraca adotado está representado na Tabela 3.2.1. Este modelo é uma versão ligeiramente modificada do modelo assintótico EKV, válido em inversão fraca, (BUCHER, 1996, Passim), no qual os parâmetros de ajuste  $a$  e  $b$ , independentes da tecnologia e adimensionais, são introduzidos de modo a melhorar a aproximação para  $i_{f(r)} < 0,6$  (SOBRAL, 2008, Passim).  $I_D$  é a corrente de dreno,  $I_S$  é a corrente específica,  $i_{f(r)}$  é a componente de saturação direta (reversa) normalizada da corrente de dreno,  $\phi_t$  é a tensão térmica,  $V_P$  é a tensão de “pinch-off”,  $V_{SB(DB)}$  é a tensão da fonte (dreno)-substrato,  $\mu$  é a mobilidade dos portadores de carga,  $C'_{ox}$  é a capacitância do óxido por unidade de área,  $n$  é fator de rampa (“slope factor”),  $W$  e  $L$  são a largura e o comprimento do canal, respectivamente, e  $V_{GB}$  é a tensão porta-substrato.

Tabela 3.2.1 — Modelo MOSFET adotado no projeto

Variável ou parâmetro	Não Saturação	Saturação
$I_D$	$b \cdot I_S (i_f - i_r)$	$b \cdot I_S \cdot i_f$
$i_{f(r)}$	$e^{\frac{(V_P - V_{SB(DB)})}{a \phi_t}}$	
$b I_S$	$2.2344 \mu C'_{ox} n \phi_t^2 \frac{W}{L}$	
$V_P$	$V_P \cong \frac{V_{GB} - V_{T0}}{n}$	
$a$	1,0382	

### 3.2.1 Problema de Otimização

O método dos Pontos Interiores foi aplicado ao projeto de um Integrador de Seevinck, de domínio logarítmico e tensão mínima, em sua versão balanceada (“fully differential”) (MANIERO, 2003, Passim), ilustrada na Fig. 3.2.



Todos os transistores na Fig. 3.2 devem operar em inversão fraca e  $M_{6A(B)}$  e  $M_{8A(B)}$  devem operar em não saturação. Os outros dispositivos MOS na Fig.3.2 devem operar em saturação. Estas condições de polarização geram as restrições (3.2.1)-(3.2.8), listadas na Tabela 3.2.2 (SOBRAL, 2008, Passim). Nesta Tabela,  $I_{BIAS}$ ,  $I$  e  $I_0$  são correntes de polarização e  $i_{INmin}$  e  $i_{INmax}$  são os valores mínimo e máximo da corrente de entrada, respectivamente. Assumindo  $(W/L)_1 = (W/L)_2$ ,  $(W/L)_7 = (W/L)_8$ , bem como as condições (3.2.8) na Tabela 3.2.2, chega-se à expressão (3.2.9) (SOBRAL, 2008, Passim) da constante de integração  $K_{INT}$  na Tabela 3.2.2. A condição  $(W/L)_6 = (W/L)_{11}$  é necessária para o cancelamento de termos indesejáveis na expressão integral relacionando a entrada diferencial de corrente à saída diferencial de corrente ( $i_{OUTB} - i_{OUTA}$ ). Finalmente, para que  $M_{7A(B)}$  não entre em corte, a restrição (3.2.10) precisa ser utilizada.

Tabela 3.2.2 — Restrições de espaço de projeto

$\frac{W_1}{L} > \left[ \left( 1 - \frac{W_3}{W_{10}} \frac{I_0}{I + I_0} \right) \left( I_{BIAS} + \frac{i_{INmax}}{2} \right) - I_0 \right] \frac{1}{0,6I_{Ssq}}$	(3.2.1)
$\frac{W_3}{L} > \frac{I_0 + I}{0,6I_{Ssq}}$	(3.2.2)
$\frac{W_7}{L} > \frac{I_{BIAS} + \frac{i_{INmax}}{2}}{0,6I_{Ssq}}$	(3.2.3)
$\frac{W_{10}}{L} > \frac{I_0}{0,6I_{Ssq}}$	(3.2.4)
$\frac{W_{11}}{L} > \frac{I_0}{0,6I_{Ssq}}$	(3.2.5)
$\frac{I}{I_0} < \frac{W_3}{W_{10}} e^{\frac{4}{a}} - 1$	(3.2.6)
$\frac{\left( I_{BIAS} + \frac{i_{INmin}}{2} \right)}{\left( \frac{W_1}{L} \right)} \left[ \left( \frac{I + I_0}{I_0} \right) \frac{W_{10}}{W_3} - 1 \right] - \left( \frac{I + I_0}{I_0} \right) \frac{W_{10}}{W_3} > b \cdot I_{Ssq} e^{\frac{4\phi_t - V_{T0}}{na\phi_t}}$	(3.2.7)
$\frac{\left( \frac{W}{L} \right)_3}{\left( \frac{W}{L} \right)_{10}} = \frac{\left( \frac{W}{L} \right)_4}{\left( \frac{W}{L} \right)_{11}}$	(3.2.8)
$\frac{1}{Ca\phi_t} \left( I + I_0 - \frac{W_3}{W_{10}} I_0 \right) = K_{INT}$	(3.2.9)
$I_{BIAS} > \frac{-i_{INmin}}{2}$	(3.2.10)

O objetivo do procedimento de otimização aqui adotado é minimizar a área de silício. Somente as áreas ativas dos transistores  $M_{1A(B)}-M_{4A(B)}$ ,  $M_{6A(B)}-M_{8A(B)}$ ,  $M_{10A(B)}$  e  $M_{11A(B)}$ , bem como as áreas dos dois capacitores foram consideradas. Para simplificar, assumiu-se o mesmo comprimento de canal para os dispositivos mencionados. Portanto, a função objetivo de nosso problema de otimização é:

$$f_o = Area = 2L(2W_1 + W_3 + W_4 + 2W_7 + W_{10} + 2W_{11}) + 2 \frac{C}{C_0} \quad (3.2.11),$$

onde  $C_0$  é a capacitância por unidade de área dos capacitores metal-óxido-metal para uma tecnologia especificada. Além da capacitância de integração e das dimensões do MOSFET que aparecem ao lado direito de (3.2.11), as outras variáveis do projeto que devem ser calculadas pela metodologia de otimização são as correntes de polarização  $I_{BIAS}$ ,  $I$  e  $I_0$ , que estão incluídas nas restrições de espaço de projeto. A constante de integração  $K_{INT}$  e os valores mínimo e máximo da corrente de entrada,  $i_{INmin}$  e  $i_{INmax}$ , são especificações do projeto, e devem ser tratadas como constantes.

### 3.2.2 Procedimentos para a Otimização

Como no problema do divisor de tensão CMOS, a otimização foi realizada usando o “Toolbox” de Otimização do MATLAB (THE MATHWORKS, 2010, Passim), aí incluída uma versão de avaliação do pacote Knitro (WALTZ, 2010, Passim), que usa métodos de Pontos Interiores. Estes métodos têm sido recomendados na literatura para casos como o do problema de Seevinck, uma vez que tratam de problemas não lineares e não convexos. Embora seja muito difícil encontrar um mínimo global, mínimos locais podem ser achados com uma quantidade de iterações relativamente pequena, a depender das tolerâncias escolhidas.

As restrições lineares são tratadas como matrizes e vetores. Nosso segundo problema não possui qualquer igualdade linear. Uma vez que o “toolbox” lida com equações e funções, faz-se necessária uma simplificação, obtida pela junção de todas as variáveis como elementos de um vetor  $x$ . Os valores iniciais para todas as variáveis, bem como seus limites (“bounds”) inferiores e superiores, devem ser definidos antes que o algoritmo da otimização seja iniciado. No próximo passo são calculados os gradientes da função objetivo e de cada restrição não linear, que são incluídos nos cálculos.

Após a escolha das opções disponíveis para as iterações, que incluem deixar o “toolbox” calcular os gradientes e hessianas, o programa é ativado e calcula a quantidade de iterações especificada até que as tolerâncias sejam atingidas. Neste problema, as hessianas foram calculadas pelo pacote Knitro, ao invés de serem incluídas.

Um relatório final é produzido mostrando os valores otimizados e um arquivo é gravado para ser aberto pelo simulador de circuitos SMASH (DOLPHIN INTEGRATION INC., 2008, Passim). Experimentações podem ser realizadas com valores iniciais diversos, na tentativa de se achar outros mínimos locais.

### 3.2.3 Resultados do projeto do Integrador de Seevinck

A ferramenta de otimização foi aplicada a três conjuntos de especificações:

$$A) K_{INT} = 2\pi \times 10^5 \text{ rad/s}, i_{INmax} = -i_{INmin} = 200 \text{ nA}$$

$$B) K_{INT} = 2\pi \times 10^5 \text{ rad/s}, i_{INmax} = -i_{INmin} = 20 \text{ nA}$$

$$C) K_{INT} = 2\pi \times 10^6 \text{ rad/s}, i_{INmax} = -i_{INmin} = 20 \text{ nA}$$

Foi adotada a tecnologia IBM 0.13, para a qual  $bI_S = 353 \text{ nA}$ ,  $V_{T0} = 0,362 \text{ V}$ ,  $n = 1,38$  e  $C_0 = 2,113 \times 10^{-3} \text{ F/m}^2$ .

A tensão total fornecida pela fonte é  $1,2 \text{ V}$ .

Os seguintes limites inferiores e superiores foram assumidos para as onze variáveis do projeto:

$$0,26 \mu\text{m} \leq (W, L) \leq 100 \mu\text{m}$$

$$1 \text{ pA} \leq (I, I_0, I_{BIAS}) \leq 10 \mu\text{A}$$

$$10 \text{ pF} \leq C \leq 100 \text{ pF}.$$

Os resultados otimizados são apresentados na Tabela 3.2.3 para os três exemplos do projeto. A última linha mostra o valor minimizado da área, expresso por (3.2.11). Nas linhas superiores,  $A_{tran}$  é a área ativa de todos os transistores na Fig. 3.1 e  $A_{cap}$  é a área dos dois capacitores.

A quantidade de iterações necessárias até a convergência foi de:

31 para o exemplo A,  
63 para o exemplo B e  
28 para o exemplo C.

Os erros calculados através da verificação dos valores das desigualdades em (3.2.8) e (3.2.9) da Tabela 3.2.2 foram menores que 0,01 % nos três exemplos.

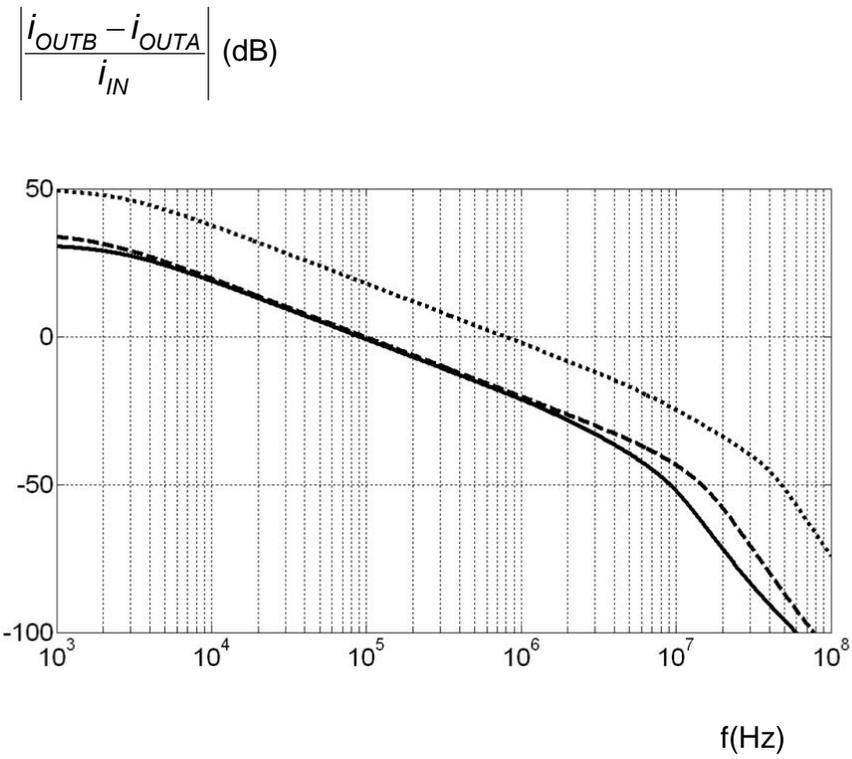
As dimensões dos transistores e do capacitor, bem como as correntes de polarização, foram carregadas no simulador de circuitos Smash (DOLPHIN INTEGRATION, 2008, Passim). As simulações das respostas em frequência de pequenos sinais estão mostradas na Fig. 3.3 para os três exemplos do projeto. Na Tabela 3.2.4, são apresentados alguns resultados de simulações correspondentes ao desempenho do circuito. Nesta tabela, a constante de integração  $K_{INT}$  é obtida a partir do diagrama de Bode de módulo da função de transferência, interpolando o valor da frequência em que o módulo é unitário (0 dB). Um pólo dominante diferente de zero é devido aos efeitos de canal curto, que produzem descasamento elétrico (“mismatching”) nesta arquitetura (MANIERO, 2003, Passim) (estes efeitos não foram levados em conta no modelo da Seção 3.1). A largura de faixa referida na segunda linha da Tabela 3.2.4 refere-se à região com inclinação de -20 dB/década. Esta largura de faixa degrada com a frequência, pois as capacitâncias parasitas dos dispositivos não foram consideradas no modelo adotado. Esta largura de faixa degrada com a frequência, pois as capacitâncias parasitas dos dispositivos não foram consideradas no modelo adotado. Pela mesma razão torna-se mais difícil obter uma concordância estreita entre os valores especificados e simulados de  $K_{INT}$  para grandes valores desta constante, tal como no exemplo C do projeto.

**Tabela 3.2.3 — Resultados Otimizados para os Exemplos A, B e C**

<b>Variáveis</b>	<b>Exemplo A</b>	<b>Exemplo B</b>	<b>Exemplo C</b>
$I_{BIAS}$ ( $\mu A$ )	5,57	3,20	7,26
$I$ ( $\mu A$ )	1,28	1,88	6,88
$I_0$ (nA)	160	547	650
$L$ ( $\mu m$ )	0,33	0,36	0,33
$W_1 = W_2$ ( $\mu m$ )	61,04	56,65	53,82
$W_3$ ( $\mu m$ )	71,26	71,19	66,39
$W_4$ ( $\mu m$ )	74,93	68,42	59,33
$W_6 = W_{11}$ ( $\mu m$ )	40,21	33,95	60,95
$W_7 = W_8$ ( $\mu m$ )	66,32	62,18	68,68
$W_{10}$ ( $\mu m$ )	38,24	35,32	68,20
$C$ (pF)	67,15	78,19	40,67
$A_{tran}$ ( $\mu m^2$ )	343	346	370
$A_{cap}$ ( $\mu m^2$ )	$63,57 \times 10^3$	$74,02 \times 10^3$	$38,50 \times 10^3$
<b>Área</b> ( $\mu m^2$ )	$63,91 \times 10^3$	$74,36 \times 10^3$	$38,87 \times 10^3$

Nota-se que a área dos capacitores  $A_{cap}$  predomina expressivamente sobre o conjunto de áreas dos transistores  $A_{tran}$  na função objetivo, sugerindo que este problema praticamente se reduz a uma minimização da área dos capacitores. Entretanto, para se obter uma constante de integração especificada, deve-se satisfazer a restrição (3.2.9) da Tabela 3.2.2, que inclui, além do valor da capacitância, correntes de polarização e dimensões de transistores, que por sua vez, são relacionadas a outras variáveis através de outras restrições do projeto.

**Figura 3.3 — Respostas em frequência de pequenos sinais do integrador de Seevinck para: Projeto A (linha sólida); Projeto B (linha tracejada); Projeto C (linha pontilhada).**



**Tabela 3.2.4 — Resultados de simulações para os exemplos A, B e C**

<b>Características</b>	<b>Exemplo A</b>	<b>Exemplo B</b>	<b>Exemplo C</b>
Frequência do polo dominante (kHz)	2,6	1,7	2,6
Largura de faixa (MHz)	1,98	4,91	7,01
$\frac{K_{INT}}{2\pi}$ (Hz)	$90,9 \times 10^4$	$98,0 \times 10^4$	$793 \times 10^5$
Potência DC ( $\mu$ W)	36,86	26,99	76,90
Intermodulação de 3ª ordem em 90 kHz (dB)*	-48,12	-35,40	-39,43

(\*) tons de entrada em 100 e 110 kHz com amplitudes de  $\frac{i_{INmax}}{2}$

### **3.2.4 Conclusões sobre o projeto do Integrador de Seevinck.**

O método dos Pontos Interiores para otimização foi aplicado com sucesso no projeto de um integrador de domínio logarítmico, que consiste em um problema não convexo. Os valores otimizados das variáveis do projeto foram testadas através de simulações e o desempenho obtido foi satisfatório, uma vez que o circuito se comporta como integrador (declividade de  $-20$  dB/década) para uma faixa de frequências expressiva e apresenta distorção muito baixa. Tal desempenho confirma a confiabilidade do procedimento de otimização. Verificou-se também que a convergência é muito rápida.

A minimização da área provou-se desafiante no projeto do integrador de domínio logarítmico de Seevinck, uma vez que a principal contribuição para a área do circuito é dada pelos capacitores e o desempenho e as restrições de polarização geralmente demandam altos valores de capacitâncias.

## 4. CONCLUSÕES FINAIS

Este trabalho consiste numa abordagem pioneira na tentativa de encontrar soluções de área mínima para projetos de circuitos integrados analógicos utilizando o método de otimização conhecido por Pontos Interiores. Tal método consiste numa alternativa de excelente aceitação na comunidade científica para o caso de problemas de otimização não convexos e não lineares, onde a aplicação de técnicas mais populares, como a Programação Convexa ou a Programação Geométrica, tornam-se proibitivas.

Foram escolhidos dois exemplos de circuitos analógicos de relevância para os projetistas: um divisor de tensões CMOS, onde os transistores são representados por um modelo válido em todo o regime de inversão (fraca, moderada e forte) e o integrador de domínio logarítmico ("log-domain") para mínima tensão de alimentação de Seevinck, onde os transistores operam em inversão fraca e sua característica corrente-tensão tem um comportamento exponencial.

O primeiro tipo de circuito foi escolhido pela sua vasta aplicação em circuitos analógicos, pela simplicidade do circuito e pelo número reduzido de expressões e de variáveis. Não obstante, a representação da relação corrente-tensão dos transistores por um modelo genérico para todo o regime de inversão, o modelo ACM ("Advanced Compact MOSFET Model"), levou à descrição do problema de otimização por relações não convexas, inviabilizando a utilização da Programação Convexa. A aplicação do Método de Pontos Interiores, por outro lado, mostrou-se bem sucedida, tendo requerido um número reduzido de iterações e tendo levado a resultados de simulação satisfatórios. As pequenas divergências entre especificação e simulação são devidas à não representação de efeitos de segunda ordem no modelo adotado e podem ser facilmente reduzidas através de ajustes empíricos nas dimensões encontradas, ou pela incorporação de mais parâmetros e termos corretivos na expressão básica da característica corrente-tensão do modelo ACM. A modelagem de efeitos de

segunda ordem (efeitos de canal curto, degradação da mobilidade com o campo elétrico transversal, entre outros), normalmente levam a expressões analíticas muito complexas. Entretanto, a aplicação do Método dos Pontos Interiores não oferece qualquer limitação quanto à forma ou complexidade da função objetivo e das restrições, consistindo em mais uma vantagem sobre as metodologias de programação convexa ou geométrica.

Deve-se salientar que a representação dos transistores por modelos quadráticos para suas características corrente-tensão possibilitaria, em geral, a utilização de programação convexa ou geométrica. Entretanto, tais modelos são válidos apenas no regime de inversão forte, e nossos resultados mostraram que, muito provavelmente, pelo menos um dos transistores fica polarizado em inversão fraca ou moderada, na solução ótima. Isto normalmente se verifica mesmo em cálculos manuais, quando não se visa a minimização da área, pois as diferenças tecnológicas entre os transistores e o fato de estarem sendo percorridos pela mesma corrente podem levar os MOSFETs canal N e canal P para níveis de inversão bem distintos entre si, para valores particulares do potencial de referência desejado.

Ainda com relação ao primeiro exemplo, como o Método de Pontos Interiores não garante a convergência para um mínimo global, mas para um mínimo local que depende das escolhas dos valores iniciais das variáveis, procedemos a uma varredura de condições iniciais através da combinação de todos os extremos, inferiores e superiores. Como se tratava de um problema com cinco variáveis, 32 foram as combinações possíveis, para as quais a rotina de otimização foi repetidamente aplicada. Acreditamos que na maior parte dos problemas esta metodologia permitirá identificar o mínimo global da função objetivo, que seria o menor dentre os mínimos locais encontrados. Embora se trate de um processo exaustivo, sobretudo se o número de variáveis do problema for grande, este inconveniente é amenizado pelo fato de rotinas de otimização segundo o método dos Pontos Interiores convergirem muito rapidamente. No caso particular do divisor de tensão CMOS, uma possível convexidade da função objetivo (mas não das restrições) conduziu à

convergência da rotina para uma mesma solução, para qualquer das 32 combinações de valores iniciais.

O segundo tipo de circuito escolhido, menos difundido na literatura, mas de crescente interesse para aplicações analógicas que requerem baixo consumo e baixa tensão, como é o caso de dispositivos bio-implantáveis ou dispositivos autônomos, é um integrador que confere alta linearidade aos filtros em que é utilizado. Contudo, seu projeto é de difícil execução, dado o número de condições a serem verificadas e a necessidade de utilização de um modelo exponencial para a relação corrente-tensão dos transistores. A programação convexa aqui também não se aplica, justificando a abordagem com o Método dos Pontos Interiores na tentativa de automatizar o projeto com algum critério de otimização. Neste exemplo, a minimização da área mostrou consistir essencialmente da minimização da área do capacitor de integração (expressivamente predominante sobre a área ativa dos transistores), o que, contudo não simplifica o problema, uma vez que o valor capacitivo, portanto sua área, relaciona-se por meio de complexas restrições de projeto às demais variáveis.

Embora não tenha sido aplicada uma varredura de combinações dos valores iniciais, como no exemplo anterior, as soluções encontradas resultaram em valores pequenos para as áreas e o desempenho dos integradores, verificado por simulação, mostrou-se muito adequado. Mais uma vez, o número de iterações requerido foi bastante baixo.

Em ambos os exemplos, adotou-se a minimização da área dos circuitos como critério de otimização, por ser este um fator determinante de custo para os “chips”. Em circuitos integrados mistos (porções analógica e digital compartilhando a mesma pastilha), não raro a parte analógica consome uma parte significativa da área de silício, embora o processamento a ser realizado por seus componentes seja bem mais simples que o da parte digital. Daí os projetistas de circuitos integrados do século XXI, quando sistemas completos “on chip” se tornaram necessidades prementes, atribuírem tanta importância à redução da área dos módulos analógicos. Entretanto, outras características

podem ser otimizadas, em combinação com a área, como a potência DC (que também deve ser mínima, em geral), o que tornará a função objetivo mais complexa e possivelmente mais distante do ideal convexo. A combinação de outras características otimizáveis na função objetivo não é, contudo, uma tarefa fácil, pois as diferentes unidades e faixas de variação das mesmas requerem a aplicação de alguma forma eficaz de normalização e, possivelmente, de fatores de ponderação para flexibilizar o projeto.

Os exemplos abordados neste trabalho demonstraram a conveniência da utilização do Método dos Pontos Interiores para a obtenção de soluções ótimas no projeto de circuitos integrados analógicos. Trabalhos futuros neste campo podem ser realizados objetivando:

- (i) Incorporar os efeitos de segunda ordem em modelos compactos e genéricos dos transistores, a fim de tornar mais exatas as soluções otimizadas obtidas pelo Método dos Pontos Interiores;
- (ii) Adicionar técnicas de inspeção do mínimo global da função objetivo;
- (iii) Adotar algoritmos de busca global que utilizem os métodos os pontos interiores para definir os mínimos locais como valores iniciais;
- (iv) Aplicar a metodologia, associada a um modelo de transistor genérico, a outras classes de circuitos integrados analógicos, como amplificadores, particularmente os operacionais.
- (v) Realimentar os resultados do SMASH como entradas para o KNITRO;
- (vi) Desenvolver um algoritmo independente, para disponibilizar a metodologia a um público maior.

## 5. REFERÊNCIAS

BALBO, A.R. et alii. Introdução aos métodos primal-dual de pontos interiores e aplicações. In: **9th Brazilian Conference on Dynamics Control and their Applications**. Serra Negra, S.P., 2010. Disponível em: <http://www.sbmac.org.br/dincon/trabalhos/PDF/shortcourses/71855.pdf>. Acesso: 5 fev. 2011.

BOYD, S.P., VANDENBERGHE, L. **Convex Programming**. USA: Cambridge University Press, 2004.

BUCHER, M., et alii. Accurate MOS modelling for analog circuit simulation using the EKV model. In: **Circuits and Systems, Proceedings of the IEEE International Symposium**, vol.4,. USA, 1996, p. 703 – 706. Disponível em: <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=3834> Acesso em: 3 jan. 2011.

BYRD, R.H., GILBERT, J.C., NOCEDAL, J., **A Trust Region Method Based on Interior Point Techniques for Nonlinear Programming**. Report OTC 96/02, Optimization Technology Center. USA, 1998. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.163...>Acesso em: 22 fev. 2011.

BYRD, R.H., HRIBAR, M.E., NOCEDAL, J. **An interior point algorithm for large scale nonlinear programming**. USA, 1998. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.39>. Acesso em: 3 jan. 2011.

BYRD, R.H., LIU, G., NOCEDAL, J. **On the local behavior of an interior point method for linear programming**. Report OTC 98/02, Optimization Technology Center. USA, 1998. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.8.3336&rep>. Acesso em: 2 jan. 2011.

CUNHA, A.I.A., SCHNEIDER, M.C., GALUP-MONTORO, C. An MOS Transistor Model for Analog Circuit Design. **IEEE J. Solid-State Circuits**, vol. 33, USA, October 1998, p. 1510-1519.

CUNHA, A.I.A., SCHNEIDER, M.C., MONTORO, C.G., CAETANO, C.D.C, MACHADO, M.B. Unambiguous extraction of threshold voltage based on the transconductance-to-current ratio. **Technical Proceedings of the 2005 Workshop on Compact Modeling**, Anaheim, USA, May 2005, p. 139-142.

CURTIS, F. **A penalty-interior-point method for large-scale nonlinear optimization.** Dept. of Industrial & System Engineering: Lehigh University. USA, 2010. Disponível em: [http://stanford.edu/~boyd/papers/l1\\_logistic\\_reg.html](http://stanford.edu/~boyd/papers/l1_logistic_reg.html) Acesso em: 2 mar 2011.

DOLPHIN INTEGRATION INC. **SMASH & Scrooge user manual.** Versão 5. France, 2008. Disponível em: <http://www.dolphin.fr>. Acesso em: 23 jan. 2011.

FORSGREN, A., GILL, P.E., WRIGHT, M.H. Interior methods for nonlinear optimization. In: **SIAM** (Society for Industrial and Applied Mathematics) **Review**, Vol. 44, No. 4, USA, 2002, p. 525-597.

GERTZ, M., NOCEDAL, J., SARTENAER, A. A starting-point strategy for nonlinear interior methods. In: **Applied Math Letters**, Vol. 17, Amsterdam: Elsevier B.V., 2004, p. 945-952.

GOULD, N., ORBAN, D., TOINT, P. Numerical methods for large-scale nonlinear optimization. In: **Acta Numerica**, Vol. 14, United Kingdom: Cambridge University Press, 2005, p. 299-361.

GRIVA, I., NASH, S., SOFER, A. **Linear and nonlinear optimization.** 2ª Ed., Philadelphia: SIAM (Society for Industrial and Applied Mathematics), 2009.

GUARDIA, L.E., MATLAB: primal-dual para programação não linear-decomposição e iterativo. In: **Seminário Brasileiro de Análise**, São Paulo: Instituto de Matemática e Estatística, USP, Edição nº. 68, Nov. 2008.

HERSHENSON, M. M., BOYD, S. P., LEE, T.H. Optimal design of a CMOS op-amp via geometric programming. In: **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, Vol. 30, nº 6, Minneapolis: IEEE, January 2001, p. 1-21.

KARMAKAR, N. A new polynomial-time algorithm for linear programming. In: **Combinatorica**, Massachusetts: János Bolyai Mathematical Society, Vol. 4, Nº 4, 1984, p. 373-395. Disponível em: <http://www.amath.washington.edu/~narc/win08/papers/karmarkar.pdf> . Acesso em 5 mar. 2011.

LUENBERGER, D.G., YE, Y. **Linear and nonlinear programming.** Ed. 3a, USA: Springer, 2008.

MANDAL, P., VISVANATHAN, V. CMOS op-amp sizing using a geometric programming formulation. In: **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, Minneapolis: IEEE, Vol. 20, January 2001, p. 22-38.

MANIERO, A., GEROSA, A. NEVIANI, A. Performance optimization in micro-power, low-voltage log-domain filters in pure CMOS technology. In: **Circuits and Systems, Proceedings of the IEEE International Symposium on Circuits and Systems**, Vol. 1, 2003, p. 1-565 – 1-568. Disponível em: <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8570> . Acesso em: 3 fev. 2011.

MANIERO, A., GEROSA, A. NEVIANI, A. A fully integrated dual-channel log-domain programmable preamplifier and filter for an implantable cardiac pacemaker. In: **Circuits and Systems I: Regular Papers, IEEE Transactions Papers**, nº 51, 2004, p. 1916-1925. Disponível em: [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=1344216](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1344216). Acesso em: 1 fev. 2011

NOCEDAL, J., WRIGHT, S., **Numerical Optimization**. 2a ed., USA: Springer, 2006.

OLIVEIROS, J., et alii. An improved and automated design tool for the optimization of CMOS OTAs using geometric programming. In: **Proceedings of the 21st Symposium on Integrated Circuits and Systems Design**, p. 146-151, 2008. Disponível em: <http://portal.acm.org/citation.cfm?id=1404371> . Acesso em: 6 fev. 2011.

SANTANA, E.P., **Uma metodologia para projeto de amplificadores operacionais de transcondutância em tecnologia CMOS auxiliado por computador**. Dissertação de Mestrado. Escola Politécnica, Departamento de Engenharia Elétrica, Universidade Federal da Bahia. Salvador, 2005.

SEEVINCK, E., et alii. CMOS translinear circuits for minimum supply voltage. In: **IEEE Transactions on Circuits and Systems – II: Analog and Digital Signal Processing**, nº 47, 2000, p. 1560-1564. Disponível em: <http://ieeexplore.ieee.org/Xplore/login.jsp?url>. Acesso em: 15 jan. 2011.

SHARMA, S. **Applied Nonlinear Programming**. New Delhi: New Age International Limited, 2006.

SILVA, R., CUNHA, A.I., MARTINEZ, L. Optimized Sizing of Analog Circuits Using the Interior-Point Method for Nonconvex Nonlinear Functions under Matlab. In: **24th Symposium on Integrated Circuits and Systems Design (SBCCI '11)**, João Pessoa, PB, Ago. 30-Set. 2, 2011. Trabalho aceito para ser apresentado no evento.

SOBRAL, V. A. L., BOMFIM, R. E. C., LIMA, R. N., Cunha, A. I. A. Systematic methodology for the design of Seevinck's CMOS log-domain integrators. In: **Proceedings of the 21st Symposium on Integrated Circuits and Systems Design (SBCCI'08)**, 2008, p. 158-163. Disponível em: <http://portal.acm.org/citation.cfm?id=1404371>. Acesso em: 6 fev. 2011.

THE MATHWORKS INC. **MATLAB optimization toolbox, user's guide**. Versão 5, 2010. Disponível em: <http://www.mathworks.com>. Acesso em: 20 jan. 2011.

VANDERBEI, R.J., SHANNO, D. **An interior point algorithm for nonconvex nonlinear programming, statistics and operations research**. New Jersey: Princeton University, SOR-97-21, 1991.

VANDERHAEGEN, J. P., BRODERSEN, R. W. Automated Design of Operational Transconductance Amplifiers Using Reversed Geometric Programming. In: **Design Automation Conference (DAC '04) Proceedings 41 st**, 2004, p. 133-138. Disponível em: [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=1322457](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1322457). Acesso em: 22 jan. 2011.

WALTZ, R.A., PLANTENGA, T.D. **KNITRO 7.0 User's Manual**. USA: Ziena Optimization Inc., 2010.

WATERLOO MAPLE INC. **MAPLE User Manual**, Versão 13. Canada, 2010. Disponível em: <http://www.maplesoft.com>. Acesso em: 25 fev. 2011.

WRIGHT, M.H., The Interior-Point Revolution in Optimization: History, Recent Developments and Lasting Consequences. In: **Bulletin of the American Mathematical Society**, Vol. 42, n<sup>o</sup> 1, 2004, p. 39-56. Disponível em: <http://www.ams.org/journals/bull/2005-42-01/S0273-0979-04-01040-7/home.html>. Acesso em: 4 fev. 2011.

# APÊNDICES

## APÊNDICE A – Cálculos simbólicos do MAPLE

---

### A.1 Projeto do divisor de tensão

Função objetivo :

$$> f := x3 \cdot x1 + x4 \cdot x2 :$$

Gradiente da função objetivo :

$$> Gfo := \text{Vector} \left( \left[ \frac{\partial}{\partial x1} f, \frac{\partial}{\partial x2} f, \frac{\partial}{\partial x3} f, \frac{\partial}{\partial x4} f, \frac{\partial}{\partial x5} f \right] \right)$$

$$Gfo := \begin{bmatrix} x3 \\ x4 \\ x1 \\ x2 \\ 0 \end{bmatrix}$$

Hessiana da função objetivo :

>

$$\text{Hessfo} := \left( \left[ \left[ \frac{\partial}{\partial x1} Gfo(1), \frac{\partial}{\partial x2} Gfo(1), \frac{\partial}{\partial x3} Gfo(1), \frac{\partial}{\partial x4} Gfo(1), \frac{\partial}{\partial x5} Gfo(1) \right], \left[ \frac{\partial}{\partial x1} Gfo(2), \frac{\partial}{\partial x2} Gfo(2), \frac{\partial}{\partial x3} Gfo(2), \frac{\partial}{\partial x4} Gfo(2), \frac{\partial}{\partial x5} Gfo(2) \right], \left[ \frac{\partial}{\partial x1} Gfo(3), \frac{\partial}{\partial x2} Gfo(3), \frac{\partial}{\partial x3} Gfo(3), \frac{\partial}{\partial x4} Gfo(3), \frac{\partial}{\partial x5} Gfo(3) \right], \left[ \frac{\partial}{\partial x1} Gfo(4), \frac{\partial}{\partial x2} Gfo(4), \frac{\partial}{\partial x3} Gfo(4), \frac{\partial}{\partial x4} Gfo(4), \frac{\partial}{\partial x5} Gfo(4) \right], \left[ \frac{\partial}{\partial x1} Gfo(5), \frac{\partial}{\partial x2} Gfo(5), \frac{\partial}{\partial x3} Gfo(5), \frac{\partial}{\partial x4} Gfo(5), \frac{\partial}{\partial x5} Gfo(5) \right] \right] \right)$$

$$> \text{Hessfo} := \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} :$$

---

Primeira igualdade ceq(1) (não-linear) :

$$\begin{aligned}
 > \text{ceq}(1) := \left( \text{phit} \cdot \left( \text{sqrt} \left( 1 + \frac{x5}{\frac{\text{IsqP} \cdot x3}{x1}} \right) - 3 + \ln \left( \text{sqrt} \left( 1 + \frac{x5}{\frac{\text{IsqP} \cdot x3}{x1}} \right) - 1 \right) \right) + \frac{(Vx - Vdd - VtoP)}{nP} \right) :
 \end{aligned}$$

Gradiente da primeira igualdade  $\text{ceq}(1)$  :

$$> \text{Gceq1} := \text{Vector} \left( \left[ \frac{\partial}{\partial x1} \text{ceq}(1), \frac{\partial}{\partial x2} \text{ceq}(1), \frac{\partial}{\partial x3} \text{ceq}(1), \frac{\partial}{\partial x4} \text{ceq}(1), \frac{\partial}{\partial x5} \text{ceq}(1) \right] \right)$$

$$\begin{aligned}
 \text{Gceq1} := & \left[ \left[ \text{phit} \left( \frac{1}{2} \frac{x5}{\sqrt{1 + \frac{x5 x1}{\text{IsqP} x3}} \text{IsqP} x3} \right) + \frac{1}{2} \frac{x5}{\sqrt{1 + \frac{x5 x1}{\text{IsqP} x3}} \text{IsqP} x3 \left( \sqrt{1 + \frac{x5 x1}{\text{IsqP} x3}} - 1 \right)} \right], \right. \\
 & \left[ 0 \right], \\
 & \left[ \text{phit} \left( -\frac{1}{2} \frac{x5 x1}{\sqrt{1 + \frac{x5 x1}{\text{IsqP} x3}} \text{IsqP} x3^2} \right) - \frac{1}{2} \frac{x5 x1}{\sqrt{1 + \frac{x5 x1}{\text{IsqP} x3}} \text{IsqP} x3^2 \left( \sqrt{1 + \frac{x5 x1}{\text{IsqP} x3}} - 1 \right)} \right], \\
 & \left[ 0 \right], \\
 & \left. \left[ \text{phit} \left( \frac{1}{2} \frac{x1}{\sqrt{1 + \frac{x5 x1}{\text{IsqP} x3}} \text{IsqP} x3} \right) + \frac{1}{2} \frac{x1}{\sqrt{1 + \frac{x5 x1}{\text{IsqP} x3}} \text{IsqP} x3 \left( \sqrt{1 + \frac{x5 x1}{\text{IsqP} x3}} - 1 \right)} \right] \right]
 \end{aligned}$$

---

Segunda igualdade  $\text{ceq}(2)$  (não – linear) :

$$\begin{aligned} > \text{ceq}(2) := & \left( -\text{phit} \cdot \left( \text{sqrt} \left( 1 + \frac{x5}{\left( \frac{\text{IsqN} \cdot x4}{x2} \right)} \right) - 3 + \ln \left( \text{sqrt} \left( 1 + \frac{x5}{\left( \frac{\text{IsqN} \cdot x4}{x2} \right)} \right) - 1 \right) \right) + \frac{(Vx - Vss - VtoN)}{nN} \right) : \end{aligned}$$

Gradiente da segunda igualdade  $\text{ceq}(2)$

$$\begin{aligned} > \text{Gceq2} := & \text{Vector} \left( \left[ \frac{\partial}{\partial x1} \text{ceq}(2), \frac{\partial}{\partial x2} \text{ceq}(2), \frac{\partial}{\partial x3} \text{ceq}(2), \right. \right. \\ & \left. \left. \frac{\partial}{\partial x4} \text{ceq}(2), \frac{\partial}{\partial x5} \text{ceq}(2) \right] \right) \end{aligned}$$

$$\begin{aligned} \text{Gceq2} := & \left[ \begin{array}{c} 0 \\ -\text{phit} \left( \frac{1}{2} \frac{x5}{\sqrt{1 + \frac{x5 x2}{\text{IsqN} x4}} \text{IsqN} x4} + \frac{1}{2} \frac{x5}{\sqrt{1 + \frac{x5 x2}{\text{IsqN} x4}} \text{IsqN} x4 \left( \sqrt{1 + \frac{x5 x2}{\text{IsqN} x4}} - 1 \right)} \right) \\ 0 \\ -\text{phit} \left( -\frac{1}{2} \frac{x5 x2}{\sqrt{1 + \frac{x5 x2}{\text{IsqN} x4}} \text{IsqN} x4^2} - \frac{1}{2} \frac{x5 x2}{\sqrt{1 + \frac{x5 x2}{\text{IsqN} x4}} \text{IsqN} x4^2 \left( \sqrt{1 + \frac{x5 x2}{\text{IsqN} x4}} - 1 \right)} \right) \\ -\text{phit} \left( \frac{1}{2} \frac{x2}{\sqrt{1 + \frac{x5 x2}{\text{IsqN} x4}} \text{IsqN} x4} + \frac{1}{2} \frac{x2}{\sqrt{1 + \frac{x5 x2}{\text{IsqN} x4}} \text{IsqN} x4 \left( \sqrt{1 + \frac{x5 x2}{\text{IsqN} x4}} - 1 \right)} \right) \end{array} \right] \end{aligned}$$

---

Primeira restrição de desigualdade  $c(1)$  (linear) :

$x1 - mvalx1 - s1 = 0$  # função de desigualdade com variável folga

$$> c(1) := x1 - mvalx1 - s1 :$$

Gradiente da primeira desigualdade  $c(1)$

$$> Gc1 := \text{Vector}\left(\left[\frac{\partial}{\partial x1}c(1), \frac{\partial}{\partial x2}c(1), \frac{\partial}{\partial x3}c(1), \frac{\partial}{\partial x4}c(1), \frac{\partial}{\partial x5}c(1)\right]\right)$$

$$Gc1 := \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Segunda restrição de desigualdade  $c(2)$  (linear) :

$$> c(2) := -x1 + Mvalx1 - s2 :$$

Gradiente da segunda desigualdade  $c(2)$

$$> Gc2 := \text{Vector}\left(\left[\frac{\partial}{\partial x1}(c(2)), \frac{\partial}{\partial x2}(c(2)), \frac{\partial}{\partial x3}(c(2)), \frac{\partial}{\partial x4}(c(2)), \frac{\partial}{\partial x5}(c(2))\right]\right)$$

$$Gc2 := \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Terceira restrição de desigualdade  $c(3)$  (linear) :

$$> c(3) := x2 - mvalx2 - s3 :$$

Gradiente da terceira desigualdade  $c(3)$

$$> Gc3 := \text{Vector}\left(\left[\frac{\partial}{\partial x1}(c(3)), \frac{\partial}{\partial x2}(c(3)), \frac{\partial}{\partial x3}(c(3)), \frac{\partial}{\partial x4}(c(3)), \frac{\partial}{\partial x5}(c(3))\right]\right)$$

$$Gc3 := \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Quarta restrição de desigualdade  $c(4)$  (linear) :

$$> c(4) := -x2 + Mvalx2 - s4 :$$

Gradiente da quarta desigualdade  $c(4)$

$$> Gc4 := \text{Vector}\left(\left[\frac{\partial}{\partial x1}(c(4)), \frac{\partial}{\partial x2}(c(4)), \frac{\partial}{\partial x3}(c(4)), \frac{\partial}{\partial x4}(c(4)), \frac{\partial}{\partial x5}(c(4))\right]\right)$$

$$Gc4 := \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$


---

Quinta restrição de desigualdade  $c(5)$  (linear) :

$$> c(5) := x3 - mvalx3 - s5 :$$

Gradiente da quinta desigualdade  $c(5)$

$$> Gc5 := \text{Vector}\left(\left[\frac{\partial}{\partial x1}(c(5)), \frac{\partial}{\partial x2}(c(5)), \frac{\partial}{\partial x3}(c(5)), \frac{\partial}{\partial x4}(c(5)), \frac{\partial}{\partial x5}(c(5))\right]\right)$$

$$Gc5 := \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$


---

Sexta restrição de desigualdade  $c(6)$  (linear) :

$$> c(6) := -x3 + Mvalx3 - s6 :$$

Gradiente da sexta desigualdade  $c(6)$

$$> Gc6 := \text{Vector}\left(\left[\frac{\partial}{\partial x1}(c(6)), \frac{\partial}{\partial x2}(c(6)), \frac{\partial}{\partial x3}(c(6)), \frac{\partial}{\partial x4}(c(6)), \frac{\partial}{\partial x5}(c(6))\right]\right)$$

$$Gc6 := \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 0 \end{bmatrix}$$


---

Sétima restrição de desigualdade  $c(7)$  (linear) :

$$> c(7) := x4 - mvalx4 - s7 :$$

Gradiente da sétima desigualdade  $c(7)$

$$> Gc7 := \text{Vector}\left(\left[\frac{\partial}{\partial x1}(c(7)), \frac{\partial}{\partial x2}(c(7)), \frac{\partial}{\partial x3}(c(7)), \frac{\partial}{\partial x4}(c(7)), \frac{\partial}{\partial x5}(c(7))\right]\right)$$

$$Gc7 := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Oitava restrição de desigualdade  $c(8)$  (linear) :

$$> c(8) := -x4 + Mvalx4 - s8 :$$

Gradiente da oitava desigualdade  $c(8)$

$$> Gc8 := \text{Vector}\left(\left[\frac{\partial}{\partial x1}(c(8)), \frac{\partial}{\partial x2}(c(8)), \frac{\partial}{\partial x3}(c(8)), \frac{\partial}{\partial x4}(c(8)), \frac{\partial}{\partial x5}(c(8))\right]\right)$$

$$Gc8 := \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix}$$

Nona restrição de desigualdade  $c(9)$  (linear) :

$$> c(9) := x5 - mvalx5 - s9 :$$

Gradiente da nona desigualdade  $c(9)$

$$> Gc9 := \text{Vector}\left(\left[\frac{\partial}{\partial x1}(c(9)), \frac{\partial}{\partial x2}(c(9)), \frac{\partial}{\partial x3}(c(9)), \frac{\partial}{\partial x4}(c(9)), \frac{\partial}{\partial x5}(c(9))\right]\right)$$

$$Gc9 := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Décima restrição de desigualdade  $c(10)$  (linear) :

$$> c(10) := -x5 + Mvalx5 - s10 :$$

Gradiente da décima desigualdade  $c(10)$

$$\text{> } Gc10 := \text{Vector}\left(\left[\frac{\partial}{\partial x1}(c(10)), \frac{\partial}{\partial x2}(c(10)), \frac{\partial}{\partial x3}(c(10)), \frac{\partial}{\partial x4}(c(10)), \frac{\partial}{\partial x5}(c(10))\right]\right)$$

$$Gc10 := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$

---

## A.2 Projeto do Integrador de Seevinck

Função objetivo :

$$> f_0 := 2 \cdot x_4 \cdot (2 \cdot x_5 + x_6 + x_7 + 2 \cdot x_9 + x_{10} + 2 \cdot x_8) + 2 \cdot x_{11} / Co$$

$$f_0 := 2x_4(2x_5 + x_6 + x_7 + 2x_9 + x_{10} + 2x_8) + \frac{2x_{11}}{Co}$$

Gradiente da função objetivo :

$$> Gf_0 := \text{Vector}\left(\left[\frac{\partial}{\partial x_1} f_0, \frac{\partial}{\partial x_2} f_0, \frac{\partial}{\partial x_3} f_0, \frac{\partial}{\partial x_4} f_0, \frac{\partial}{\partial x_5} f_0, \frac{\partial}{\partial x_6} f_0, \frac{\partial}{\partial x_7} f_0, \frac{\partial}{\partial x_8} f_0, \frac{\partial}{\partial x_9} f_0, \frac{\partial}{\partial x_{10}} f_0, \frac{\partial}{\partial x_{11}} f_0\right]\right)$$

$$Gf_0 := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 4x_5 + 2x_6 + 2x_7 + 4x_9 + 2x_{10} + 4x_8 \\ 4x_4 \\ 2x_4 \\ 2x_4 \\ 4x_4 \\ 4x_4 \\ 2x_4 \\ \frac{2}{Co} \end{bmatrix}$$

Primeira igualdade ceq1 :

$$\frac{x_6}{x_{10}} - \frac{x_7}{x_8} \text{ (não - linear) :}$$

$$> ceq1 := \frac{x_6}{x_{10}} - \frac{x_7}{x_8}$$

$$ceq1 := \frac{x_6}{x_{10}} - \frac{x_7}{x_8}$$

Gradiente da primeira igualdade ceq1 :

$$> Gceq1 := \text{Vector}\left(\left[\frac{\partial}{\partial x_1} ceq1, \frac{\partial}{\partial x_2} ceq1, \frac{\partial}{\partial x_3} ceq1, \frac{\partial}{\partial x_4} ceq1, \frac{\partial}{\partial x_5} ceq1, \frac{\partial}{\partial x_6} ceq1, \frac{\partial}{\partial x_7} ceq1, \frac{\partial}{\partial x_8} ceq1, \frac{\partial}{\partial x_9} ceq1, \frac{\partial}{\partial x_{10}} ceq1, \frac{\partial}{\partial x_{11}} ceq1\right]\right)$$

$$G_{ceq1} := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{x_{10}} \\ -\frac{1}{x_8} \\ \frac{x_7}{x_8^2} \\ 0 \\ -\frac{x_6}{x_{10}^2} \\ 0 \end{bmatrix}$$

---

Segunda igualdade  $ceq2 = (1/(x_{11} * a * \text{phit})) * (x_1 + x_2 - (x_3/x_{10}) * x_2) - Kint$  (não - linear) :

$$> ceq2 := (1/(x_{11} * a * \text{phit})) * (x_1 + x_2 - (x_6/x_{10}) * x_2) - Kint$$

$$ceq2 := \frac{x_1 + x_2 - \frac{x_6 x_2}{x_{10}}}{x_{11} a \text{phit}} - Kint$$

Gradiente da segunda igualdade  $ceq2$ :

$$> G_{ceq2} := \text{Vector} \left( \left[ \frac{\partial}{\partial x_1} ceq2, \frac{\partial}{\partial x_2} ceq2, \frac{\partial}{\partial x_3} ceq2, \frac{\partial}{\partial x_4} ceq2, \frac{\partial}{\partial x_5} ceq2, \frac{\partial}{\partial x_6} ceq2, \frac{\partial}{\partial x_7} ceq2, \frac{\partial}{\partial x_8} ceq2, \frac{\partial}{\partial x_9} ceq2, \frac{\partial}{\partial x_{10}} ceq2, \frac{\partial}{\partial x_{11}} ceq2 \right] \right)$$

$$G_{ceq2} := \begin{bmatrix} \frac{1}{x11 \text{ a phit}} \\ \frac{1 - \frac{x6}{x10}}{x11 \text{ a phit}} \\ 0 \\ 0 \\ 0 \\ -\frac{x2}{x11 \text{ a phit} x10} \\ 0 \\ 0 \\ 0 \\ \frac{x6 x2}{x11 \text{ a phit} x10^2} \\ \frac{x1 + x2 - \frac{x6 x2}{x10}}{x11^2 \text{ a phit}} \end{bmatrix}$$

---

Primeira desigualdade  $c1 = (x1/x2) - ((x6/x10) * \exp(4/a)) + 1$   
 + 1 (não linear)

$$> c1 := (x1/x2) - ((x6/x10) * \exp(4/a)) + 1$$

$$c1 := \frac{x1}{x2} - \frac{x6 e^{\frac{4}{a}}}{x10} + 1$$

Gradiente da primeira desigualdade  $c1$  :

$$> Gc1 := \text{Vector} \left( \left[ \frac{\partial}{\partial x1} c1, \frac{\partial}{\partial x2} c1, \frac{\partial}{\partial x3} c1, \frac{\partial}{\partial x4} c1, \frac{\partial}{\partial x5} c1, \frac{\partial}{\partial x6} c1, \frac{\partial}{\partial x7} c1, \frac{\partial}{\partial x8} c1, \frac{\partial}{\partial x9} c1, \frac{\partial}{\partial x10} c1, \frac{\partial}{\partial x11} c1 \right] \right)$$

$$Gc1 := \begin{bmatrix} \frac{1}{x2} \\ -\frac{x1}{x2^2} \\ 0 \\ 0 \\ 0 \\ -\frac{4}{x10} e^a \\ 0 \\ 0 \\ 0 \\ \frac{4}{x6} e^a \\ \frac{x6 e^a}{x10^2} \\ 0 \end{bmatrix}$$

Segunda desigualdade  $c2$  :  $-\left( (x3 + iinmin/2) / \left( \frac{x5}{x4} \right) * ((x1 + x2) / x2) * (x10/x6) - 1 \right) + \left( (x1 + x2) / (x5/x4) \cdot \left( \frac{x10}{x6} \right) \right) + b \cdot Isq \cdot \exp((4 * phit - Vto) / (n * a * phit))$  (não - linear) :

$$\begin{aligned} > c2 := -\left( (x3 + iinmin/2) / \left( \frac{x5}{x4} \right) * ((x1 + x2) / x2) * (x10/x6) - 1 \right) \\ &+ \left( (x1 + x2) / (x5/x4) \cdot \left( \frac{x10}{x6} \right) \right) + b \cdot Isq \cdot \exp((4 * phit - Vto) / (n * a * phit)) \end{aligned}$$

$$\begin{aligned} c2 := & -\frac{\left( x3 + \frac{1}{2} iinmin \right) x4 \left( \frac{(x1 + x2) x10}{x2 x6} - 1 \right)}{x5} \\ & + \frac{(x1 + x2) x4 x10}{x5 x6} + b Isq e^{\frac{4 phit - Vto}{n a phit}} \end{aligned}$$

Gradiente da segunda desigualdade  $c2$  :

$$\begin{aligned} > Gc2 := & Vector \left( \left[ \frac{\partial}{\partial x1} c2, \frac{\partial}{\partial x2} c2, \frac{\partial}{\partial x3} c2, \frac{\partial}{\partial x4} c2, \frac{\partial}{\partial x5} c2, \right. \right. \\ & \left. \left. \frac{\partial}{\partial x6} c2, \frac{\partial}{\partial x7} c2, \frac{\partial}{\partial x8} c2, \frac{\partial}{\partial x9} c2, \frac{\partial}{\partial x10} c2, \frac{\partial}{\partial x11} c2 \right] \right) \end{aligned}$$

$Gc2 :=$

$$\begin{aligned}
 & \left[ \left[ -\frac{\left(x^3 + \frac{1}{2} iinmin\right) x^4 x^{10}}{x^5 x^2 x^6} + \frac{x^4 x^{10}}{x^5 x^6} \right], \right. \\
 & \left[ -\frac{\left(x^3 + \frac{1}{2} iinmin\right) x^4 \left(\frac{x^{10}}{x^2 x^6} - \frac{(x^1 + x^2) x^{10}}{x^2 x^6}\right)}{x^5} + \frac{x^4 x^{10}}{x^5 x^6} \right], \\
 & \left. \left[ -\frac{x^4 \left(\frac{(x^1 + x^2) x^{10}}{x^2 x^6} - 1\right)}{x^5} \right], \right. \\
 & \left[ -\frac{\left(x^3 + \frac{1}{2} iinmin\right) \left(\frac{(x^1 + x^2) x^{10}}{x^2 x^6} - 1\right)}{x^5} + \frac{(x^1 + x^2) x^{10}}{x^5 x^6} \right], \\
 & \left. \left[ \frac{\left(x^3 + \frac{1}{2} iinmin\right) x^4 \left(\frac{(x^1 + x^2) x^{10}}{x^2 x^6} - 1\right)}{x^5 x^2} \right. \right. \\
 & \left. \left. - \frac{(x^1 + x^2) x^4 x^{10}}{x^5 x^2 x^6} \right], \right. \\
 & \left[ \frac{\left(x^3 + \frac{1}{2} iinmin\right) x^4 (x^1 + x^2) x^{10}}{x^5 x^2 x^6 x^2} - \frac{(x^1 + x^2) x^4 x^{10}}{x^5 x^6 x^2} \right], \\
 & \left[ 0 \right], \\
 & \left[ 0 \right], \\
 & \left[ 0 \right], \\
 & \left[ -\frac{\left(x^3 + \frac{1}{2} iinmin\right) x^4 (x^1 + x^2)}{x^5 x^2 x^6} + \frac{(x^1 + x^2) x^4}{x^5 x^6} \right], \\
 & \left[ 0 \right]
 \end{aligned}$$

-----

Terceira desigualdade  $c3$ :  $-\frac{x^5}{x^4} + ((1 - (x6/x10) * (x2/(x1 + x2)))$   
 $*(x3 + iinmax/2) - x2) * (1/(0.6 * Isq))$  (não - linear) :

$$> c3 := -\frac{x^5}{x^4} + ((1 - (x6/x10) * (x2/(x1 + x2))) * (x3 + iinmax/2) - x2) * (1/(0.6 * Isq))$$

$$c3 := -\frac{x^5}{x^4} + \frac{1.667 \left( \left( 1 - \frac{x6 x2}{x10 (x1 + x2)} \right) \left( x3 + \frac{1}{2} iinmax \right) - x2 \right)}{Isq}$$

Gradiente da terceira desigualdade  $c3$  :

$$> Gc3 := \begin{bmatrix} \frac{\partial}{\partial x1} c3 \\ \frac{\partial}{\partial x2} c3 \\ \frac{\partial}{\partial x3} c3 \\ \frac{\partial}{\partial x4} c3 \\ \frac{\partial}{\partial x5} c3 \\ \frac{\partial}{\partial x6} c3 \\ \frac{\partial}{\partial x7} c3 \\ \frac{\partial}{\partial x8} c3 \\ \frac{\partial}{\partial x9} c3 \\ \frac{\partial}{\partial x10} c3 \\ \frac{\partial}{\partial x11} c3 \end{bmatrix}$$

$$\begin{aligned}
Gc3 := & \left[ \left[ \frac{1.667 x6 x2 \left( x3 + \frac{1}{2} iinmax \right)}{x10 (x1 + x2)^2 Isq} \right] \right. \\
& \left[ \frac{1}{Isq} \left( 1.667 \left( \left( -\frac{x6}{x10 (x1 + x2)} + \frac{x6 x2}{x10 (x1 + x2)^2} \right) \left( x3 \right. \right. \right. \right. \\
& \left. \left. \left. + \frac{1}{2} iinmax \right) - 1 \right) \right) \right], \\
& \left[ \frac{1.667 \left( 1 - \frac{x6 x2}{x10 (x1 + x2)} \right)}{Isq} \right], \\
& \left[ \frac{x5}{x4^2} \right], \\
& \left[ -\frac{1}{x4} \right], \\
& \left[ -\frac{1.667 x2 \left( x3 + \frac{1}{2} iinmax \right)}{x10 (x1 + x2) Isq} \right], \\
& \left[ 0 \right], \\
& \left[ 0 \right], \\
& \left[ 0 \right], \\
& \left[ \frac{1.667 x6 x2 \left( x3 + \frac{1}{2} iinmax \right)}{x10^2 (x1 + x2) Isq} \right], \\
& \left[ 0 \right]
\end{aligned}$$


---

Quarta desigualdade  $c4$ :  $-\frac{x6}{x4} + (x1 + x2)/(0.6*Isq)$  (não  $>$   
 – linear) :

$$c4 := -\frac{x6}{x4} + (x1 + x2)/(0.6*Isq)$$

$$c4 := -\frac{x6}{x4} + \frac{1.667(x1 + x2)}{Isq}$$

Gradiente da quarta desigualdade  $c4$  :

$$Gc4 := \begin{bmatrix} \frac{\partial}{\partial x1} c4 \\ \frac{\partial}{\partial x2} c4 \\ \frac{\partial}{\partial x3} c4 \\ \frac{\partial}{\partial x4} c4 \\ \frac{\partial}{\partial x5} c4 \\ \frac{\partial}{\partial x6} c4 \\ \frac{\partial}{\partial x7} c4 \\ \frac{\partial}{\partial x8} c4 \\ \frac{\partial}{\partial x9} c4 \\ \frac{\partial}{\partial x10} c4 \\ \frac{\partial}{\partial x11} c4 \end{bmatrix}$$

$$Gc4 := \begin{bmatrix} \frac{1.667}{Isq} \\ \frac{1.667}{Isq} \\ 0 \\ \frac{x6}{x4^2} \\ 0 \\ -\frac{1}{x4} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Quinta desigualdade  $c5$ :  $-\frac{x9}{x4} + (x3 + iinmax/2)/(0.6*Isq)$  (não - linear):

$$> c5 := -\frac{x9}{x4} + (x3 + iinmax/2)/(0.6*Isq)$$

$$c5 := -\frac{x9}{x4} + \frac{1.667 \left( x3 + \frac{1}{2} iinmax \right)}{Isq}$$

Gradiente da quinta desigualdade  $c5$ :

$$> Gc5 := \text{Vector} \left( \left[ \frac{\partial}{\partial x1} c5, \frac{\partial}{\partial x2} c5, \frac{\partial}{\partial x3} c5, \frac{\partial}{\partial x4} c5, \frac{\partial}{\partial x5} c5, \frac{\partial}{\partial x6} c5, \frac{\partial}{\partial x7} c5, \frac{\partial}{\partial x8} c5, \frac{\partial}{\partial x9} c5, \frac{\partial}{\partial x10} c5, \frac{\partial}{\partial x11} c5 \right] \right)$$

$$Gc5 := \begin{bmatrix} 0 \\ 0 \\ \frac{1.667}{Isq} \\ \frac{x9}{x4^2} \\ 0 \\ 0 \\ 0 \\ 0 \\ -\frac{1}{x4} \\ 0 \\ 0 \end{bmatrix}$$

> Sexta desigualdade  $c6$  :  $-\frac{x10}{x4} + x2/(0.6*Isq)$  (não — linear) :

$$> c6 := -\frac{x10}{x4} + x2/(0.6*Isq)$$

$$c6 := -\frac{x10}{x4} + \frac{1.667x2}{Isq}$$

Gradiente da sexta desigualdade  $c6$  :

$$> Gc6 := \text{Vector} \left( \left[ \frac{\partial}{\partial x1} c6, \frac{\partial}{\partial x2} c6, \frac{\partial}{\partial x3} c6, \frac{\partial}{\partial x4} c6, \frac{\partial}{\partial x5} c6, \frac{\partial}{\partial x6} c6, \frac{\partial}{\partial x7} c6, \frac{\partial}{\partial x8} c6, \frac{\partial}{\partial x9} c6, \frac{\partial}{\partial x10} c6, \frac{\partial}{\partial x11} c6 \right] \right)$$

$$Gc6 := \begin{bmatrix} 0 \\ \frac{1.667}{Isq} \\ 0 \\ \frac{x10}{x4^2} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -\frac{1}{x4} \\ 0 \end{bmatrix}$$

Sétima desigualdade  $c7$ :  $-\frac{x8}{x4} + x2/(0.6*Isq)$  (não-linear):

$$> c7 := -\frac{x8}{x4} + x2/(0.6*Isq)$$

$$c7 := -\frac{x8}{x4} + \frac{1.667x2}{Isq}$$

Gradiente da sexta desigualdade  $c7$ :

$$> Gc7 := \text{Vector}\left(\left[\frac{\partial}{\partial x1}c7, \frac{\partial}{\partial x2}c7, \frac{\partial}{\partial x3}c7, \frac{\partial}{\partial x4}c7, \frac{\partial}{\partial x5}c7, \frac{\partial}{\partial x6}c7, \frac{\partial}{\partial x7}c7, \frac{\partial}{\partial x8}c7, \frac{\partial}{\partial x9}c7, \frac{\partial}{\partial x10}c7, \frac{\partial}{\partial x11}c7\right]\right)$$

$$Gc7 := \begin{bmatrix} 0 \\ \frac{1.667}{Isq} \\ 0 \\ \frac{x8}{x4^2} \\ 0 \\ 0 \\ 0 \\ -\frac{1}{x4} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$


---

## APÊNDICE B – Rotinas do Toolbox do MATLAB

### B.1 Projeto do divisor de tensão

---

```

function [d]=acmarea(~)
% Otimização do Circuito ACM
clc; clear all;
display('OTIMIZAÇÃO DE CIRCUITOS INTEGRADOS - MATLAB OPTIMIZATION
TOOLBOX - KNITRO');
display('MÉTODOS DOS PONTOS INTERIORES');
display(' ');
% Valores iniciais sugeridos de Vx para cada conjunto de valores
iniciais de L1, L2, W1 e W2.
% -0.4, -0.25, -0.1, 0.1, 0.25, 0.4 (Volts)
Vx=input('Digite o valor de "Vx" (em Volts): \n ? ');
NIter=input('Defina a quantidade máxima de iterações do KNITRO: \n ?
');
if NIter <=0
    display('Erro');return
end
% Tolerâncias
tolx=1e-15; % tolerância para as variáveis
tolfun=1e-06; % optimality error (função objetivo)
tolcon=1e-06; % tolerância para a região factível das restrições
fprintf('\nOs valores "default" das tolerâncias são:\nTolX= 1e-15
(variáveis)\nTolFun=1e-06 (função objetivo)\nTolCon=1e-06 (região
factível das restrições)');
fprintf('\n\nEscolha se quer mudar as tolerâncias.\n');
rel=input('Digite "S" ou "N": ','s');
if strcmp(rel,'S')==1
    tolx=input('Defina a tolerância das variáveis (TolX): \n? ');
    if tolx <=0
        display('Erro');return
    end
    tolfun=input('Defina a tolerância da função objetivo (TolFun): \n?
');
    if tolfun <=0
        display('Erro');return
    end
    tolcon=input('Defina a tolerância das restrições (TolCon): \n? ');
    if tolcon <=0
        display('Erro');return
    end
elseif strcmp(rel,'N')==1
else display('Erro');return
end
display(' ');
display('=====
=====');
fprintf('Cálculo da Otimização com o MATLAB Optimization Toolbox/KNITRO
\n');
disp('Método da Otimização: Pontos Interiores');

% Restrições limites ("bounds") das variáveis

```

```

% L1(m) L2(m) W1(m) W2(m) Id (A)
lb=[2.7e-7; 2.7e-7; 2.7e-7; 2.7e-7; 1e-12]; % limites inferiores
% L1(m) L2(m) W1(m) W2(m) Id(A)
ub=[1e-4; 1e-4; 1e-4; 1e-4; 1e-4]; % limites superiores
fprintf('\nLimites ("bounds") inferiores e superiores:\n%1.3s <= L1
<= %2.3s\n%1.3s <= L2 <= %2.3s\n%1.3s <= W1 <= %2.3s\n%1.3s <= W2
<= %3.3s\n%1.3s <= Id <=
%3.3s\n',lb(1),ub(1),lb(2),ub(2),lb(3),ub(3),lb(4),ub(4),lb(5),ub(5));
fprintf('\nTolerâncias especificadas:\nTolX= %1.0e
(variáveis)\nTolFun=%1.0e (função objetivo)\nTolCon=%1.0e (região
factível das restrições)\n',tolx,tolfun,tolcon);
display(' ');
display('=====
=====');
display('=====
=====');
% Valores iniciais
% Constantes
Vss=-0.6; % Volts
Vdd=0.6; % Volts
phit= 0.26e-1;
IsqN= 353.38*10^(-9); % Amperes
IsqP=49.514*10^(-9); % Amperes
VtoN=0.3618; % Volts
VtoP=-0.3554; % Volts
nN=1.3790;
nP=1.2952;

% Igualdades lineares (coeficientes)
Aeq = [];
beq = [];
% Desigualdades lineares (coeficientes)
A=[];
b=[];

%=====
=====
% Valores iniciais sugeridos de Vx para cada conjunto de valores
iniciais de L1, L2, W1 e W2.
% -0.4, -0.25, -0.1, 0.1, 0.25, 0.4 (Volts)

% Valores iniciais para L1, L2, W1 e W2:
% L1 L2 W1 W2 Id
Valini=[lb(1) lb(2) lb(3) lb(4) lb(5)
ub(1) lb(2) lb(3) lb(4) lb(5)
lb(1) ub(2) lb(3) lb(4) lb(5)
ub(1) ub(2) lb(3) lb(4) lb(5)
lb(1) lb(2) ub(3) lb(4) lb(5)
ub(1) lb(2) ub(3) lb(4) lb(5)
lb(1) ub(2) ub(3) lb(4) lb(5)
ub(1) ub(2) ub(3) lb(4) lb(5)
lb(1) lb(2) lb(3) ub(4) lb(5)
ub(1) lb(2) lb(3) ub(4) lb(5)
lb(1) ub(2) lb(3) ub(4) lb(5)
ub(1) ub(2) lb(3) ub(4) lb(5)
lb(1) lb(2) ub(3) ub(4) lb(5)

```

```

ub(1) lb(2) ub(3) ub(4) lb(5)
lb(1) ub(2) ub(3) ub(4) lb(5)
ub(1) ub(2) ub(3) ub(4) lb(5)
lb(1) lb(2) lb(3) lb(4) ub(5)
ub(1) lb(2) lb(3) lb(4) ub(5)
lb(1) ub(2) lb(3) lb(4) ub(5)
ub(1) ub(2) lb(3) lb(4) ub(5)
lb(1) lb(2) ub(3) lb(4) ub(5)
ub(1) lb(2) ub(3) lb(4) ub(5)
lb(1) ub(2) ub(3) lb(4) ub(5)
ub(1) lb(2) lb(3) ub(4) ub(5)
ub(1) lb(2) lb(3) ub(4) ub(5)
lb(1) ub(2) lb(3) ub(4) ub(5)
ub(1) ub(2) lb(3) ub(4) ub(5)
lb(1) lb(2) ub(3) ub(4) ub(5)
ub(1) lb(2) ub(3) ub(4) ub(5)
lb(1) ub(2) ub(3) ub(4) ub(5)
ub(1) ub(2) ub(3) ub(4) ub(5)];
NK=0; % contador de exemplos
for K=1:length(Valini) % 32 linhas na matriz Valini
    x0=[Valini(K,1); Valini(K,2); Valini(K,3); Valini(K,4);
Valini(K,5)];
    % Mostra os dados de entrada fornecidos
NK=NK+1; % incrementa o contador de exemplos
fprintf('EXEMPLO NÚMERO %u\n',NK');
display('Dados de entrada:');
display(['Vx=',num2str(Vx),' Volts']);
display(' ');
fprintf('Valores iniciais das
variáveis:\nL1=%3.5s\nL2=%3.5s\nW1=%3.5s\nW2=%3.5s\nId=%3.5s\n',x0(1),
x0(2),x0(3),x0(4),x0(5));
display('=====
=====');
display('Resultado das iterações:');
% Calcula iterações e gera relatório detalhado
[x1,x2,x3,x4,x5,Vx]=iterotima(x0,Vx,phit,IsqN,IsqP,VtoN,VtoP,nN,nP,Vss
,Vdd,lb,ub,Aeq,beq,A,b,NIter,tolx,tolfun,tolcon);
textocir=acsmash(x1,x2,x3,x4,Vdd,Vss); % Grava arquivo para a
simulação do circuito no SMASH;
% Verificação dos resultados finais
verifica=testa(x1,x2,x3,x4,x5,Vx,phit,IsqN,IsqP,VtoN,VtoP,nN,nP,Vss,Vd
d,lb,ub);
end % for K
end % FIM DO PROGRAMA

% Esta função calcula a otimização e gera um relatório das iterações
function
[x1,x2,x3,x4,x5,Vx]=iterotima(x0,Vx,phit,IsqN,IsqP,VtoN,VtoP,nN,nP,Vss
,Vdd,lb,ub,Aeq,beq,A,b,NIter,tolx,tolfun,tolcon)
% Opções do Toolbox "knitro" (Obs.: as opções de Gradientes da Função
Objetivo e Restrições devem ser as mesmas)
options =
optimset('AlwaysHonorConstraints','bounds','Algorithm','interior-
point','InitBarrierParam',0.1,'Display','final','GradObj','on','GradCo

```

```

nstr','on','Hessian','bfgs','MaxIter',NIter,'TolX',tolx,'TolFun',tolfun,
n,'TolCon',tolcon);
% Chama KNITRO para resolver o modelo de otimização.
[x]=ktrlink(@objfun,x0,A,b,Aeq,beq,lb,ub,@constfun,options);
x1=x(1);x2=x(2);x3=x(3);x4=x(4);x5=x(5);

% Função objetivo
function [f,g] = objfun(x)
    f=(x(3)*x(1)+ x(4)*x(2));
    %Gradiente da função objetivo
    g=[x(3);x(4);x(1);x(2);0];
end % função objfun

% Restrições
function [c,ceq,Gc,Gceq]= constfun(x)
% Funções de restrições não lineares 'c' (Desigualdades '<=0')
c=[];
% Gradiente das desigualdades não lineares 'c'
Gc=[];
% Funções de restrições não lineares (Igualdades '=0')
ceq=[phit*(sqrt(1+x(5)*x(1)/(IsqP*x(3)))-
3+log(sqrt(1+x(5)*x(1)/(IsqP*x(3)))-1)+(Vx-Vdd-VtoP)/nP
      (Vx-Vss-VtoN)/nN-phit*(sqrt(1+x(5)*x(2)/(IsqN*x(4)))-
3+log(sqrt(1+x(5)*x(2)/(IsqN*x(4)))-1))];
% Gradiente das igualdades não lineares 'ceq'
Gceq1=[phit*((1/2)*x(5)/(sqrt(1+x(5)*x(1)/(IsqP*x(3)))*IsqP*x(3))+(1/2)
)*x(5)/(sqrt(1+x(5)*x(1)/(IsqP*x(3)))*IsqP*x(3)*(sqrt(1+x(5)*x(1)/(Isq
P*x(3)))-1)))
      0
      phit*(-
(1/2)*x(5)*x(1)/(sqrt(1+x(5)*x(1)/(IsqP*x(3)))*IsqP*(x(3)^2))-
(1/2)*x(5)*x(1)/(sqrt(1+x(5)*x(1)/(IsqP*x(3)))*IsqP*(x(3)^2)*(sqrt(1+x
(5)*x(1)/(IsqP*x(3)))-1)))
      0

phit*((1/2)*x(1)/(sqrt(1+x(5)*x(1)/(IsqP*x(3)))*IsqP*x(3))+(1/2)*x(1)/
(sqrt(1+x(5)*x(1)/(IsqP*x(3)))*IsqP*x(3)*(sqrt(1+x(5)*x(1)/(IsqP*x(3))
)-1)))];
Gceq2=[0
      -
phit*((1/2)*x(5)/(sqrt(1+x(5)*x(2)/(IsqN*x(4)))*IsqN*x(4))+(1/2)*x(5)/
(sqrt(1+x(5)*x(2)/(IsqN*x(4)))*IsqN*x(4)*(sqrt(1+x(5)*x(2)/(IsqN*x(4))
)-1)))
      0
      -phit*(-
(1/2)*x(5)*x(2)/(sqrt(1+x(5)*x(2)/(IsqN*x(4)))*IsqN*(x(4)^2))-
(1/2)*x(5)*x(2)/(sqrt(1+x(5)*x(2)/(IsqN*x(4)))*IsqN*(x(4)^2)*(sqrt(1+x
(5)*x(2)/(IsqN*x(4)))-1)))
      -
phit*((1/2)*x(2)/(sqrt(1+x(5)*x(2)/(IsqN*x(4)))*IsqN*x(4))+(1/2)*x(2)/
(sqrt(1+x(5)*x(2)/(IsqN*x(4)))*IsqN*x(4)*(sqrt(1+x(5)*x(2)/(IsqN*x(4))
)-1)))];
Gceq=[Gceq1, Gceq2];
end % função constfun
end % função iterotima

```

```

% Esta função grava um arquivo do circuito otimizado para ser usado no
Smash
function [textocir]=acmsmash(x1,x2,x3,x4,Vdd,Vss)
textocir='ACM.cir'; % nome do arquivo de descrição do circuito
fid = fopen(textocir,'w'); % cria arquivo-texto para o SMASH
fprintf(fid,'* Arquivo de simulação no SMASH (baseado em ideia de
Victor Ariel-UFBA)\n');
fprintf(fid,'* Unidades: metro, Volt, Ampere, Farad, Ohm \n');
fprintf(fid,'* Divisor de Tensão - Modelo ACM \n');
fprintf(fid,'* %s \r\n',datestr(now,'dd/mm/yyyy HH:MM:SS \n'));
fprintf(fid,'M1 OUT OUT DD DD P W=%e L=%e \n', x3, x1);
fprintf(fid,'M2 OUT OUT SS SS N W=%e L=%e \n\n', x4, x2);
fprintf(fid,'VDD DD 0 DC %eV \n',Vdd);
fprintf(fid,'VSS SS 0 DC %eV \n\n',Vss);
fprintf(fid,'.AC DEC 100 100 100MEG \n'); % 100 pontos por década /
entre 100 Hz e 100 MHz
fprintf(fid,'.TRAN 100ns 2000us 1000us \n');
fprintf(fid,'.Trace AC {IOUTDB = DB((I(VOUTB) - I(VOUTA))/2)}
\n');
fprintf(fid,'.Trace TRAN {IOUT = I(VOUTB) - I(VOUTA)} \n');
fprintf(fid,'.Trace TRAN I(IINA1) I(IINA2) \n');
fprintf(fid,'.CAPAMIN 1f \n\n');
fprintf(fid,'* Dados da Tecnologia \n');
fprintf(fid,'* BSIM3v3.1 parameters for IBM 0.13\n');
fprintf(fid,'* Technology: SCN013\n');
fprintf(fid,'*RUN: T93V (#2_SRF_8LM_DM)\n');
fprintf(fid,'*SPICE 3f5 Level 8, Star-HSPICE Level 49, UTMOST Level
8\n');
fprintf(fid,'.MODEL N NMOS ( LEVEL =
8\n');
fprintf(fid,'+VERSION = 3.1 TNOM = 27 TOX
= 3.2E-9\n');
fprintf(fid,'+XJ = 1E-7 NCH = 2.3549E17 VTH0
= 0.0423138\n');
fprintf(fid,'+K1 = 0.3322485 K2 = -0.0298343 K3
= 1E-3\n');
fprintf(fid,'+K3B = 3.6987372 W0 = 1E-7 NLX
= 1E-6\n');
fprintf(fid,'+DVTOW = 0 DVT1W = 0 DVT2W
= 0\n');
fprintf(fid,'+DVT0 = 1.2848863 DVT1 = 0.1532141 DVT2
= 0.2740377\n');
fprintf(fid,'+U0 = 446.4279855 UA = -1.40947E-10 UB
= 3.152185E-18\n');
fprintf(fid,'+UC = 4.34007E-10 VSAT = 7.64383E4 A0
= 0.1112707\n');
fprintf(fid,'+AGS = 0.0734135 B0 = 3.913953E-6 B1
= 5E-6\n');
fprintf(fid,'+KETA = 0.05 A1 = 1.042383E-3 A2
= 0.7518921\n');
fprintf(fid,'+RDSW = 150 PRWG = 0.3496099 PRWB
= 0.1112769\n');
fprintf(fid,'+WR = 1 WINT = 2.378741E-10 LINT
= 1.040187E-8\n');
fprintf(fid,'+DWG = 1.464878E-8 DWB = 9.306917E-9 VOFF
= -0.0240657\n');

```

```

fprintf(fid,'+NFACTOR = 2.5          CIT      = 0          CDSC
= 2.4E-4\n');
fprintf(fid,'+CDSCD   = 0          CDSCB   = 0          ETA0
= 2.750169E-6\n');
fprintf(fid,'+ETAB    = -0.0151402 DSUB    = 4.056696E-6 PCLM
= 1.9784063\n');
fprintf(fid,'+PDIBLC1 = 0.9657497   PDIBLC2 = 0.01
PDIBLCB = 0.1\n');
fprintf(fid,'+DROUT   = 0.9990298   PSCBE1  = 7.957109E10 PSCBE2
= 5.015981E-10\n');
fprintf(fid,'+PVAG    = 0.535376    DELTA   = 0.01      RSH
= 6.4\n');
fprintf(fid,'+MOBMOD  = 1          PRT     = 0          UTE
= -1.5\n');
fprintf(fid,'+KT1     = -0.11       KT1L    = 0          KT2
= 0.022\n');
fprintf(fid,'+UA1     = 4.31E-9     UB1     = -7.61E-18  UC1
= -5.6E-11\n');
fprintf(fid,'+AT      = 3.3E4       WL      = 0          WLN
= 1\n');
fprintf(fid,'+WW      = 0          WWN     = 1          WWL
= 0\n');
fprintf(fid,'+LL      = 0          LLN     = 1          LW
= 0\n');
fprintf(fid,'+LWN     = 1          LWL     = 0          CAPMOD
= 2\n');
fprintf(fid,'+XPART   = 0.5         CGDO    = 3.74E-10   CGSO
= 3.74E-10\n');
fprintf(fid,'+CGBO    = 1E-12      CJ      = 9.581155E-4  PB
= 0.9756037\n');
fprintf(fid,'+MJ      = 0.4044076   CJSW    = 1E-10     PBSW
= 0.8002027\n');
fprintf(fid,'+MJSW    = 0.6         CJSWG   = 3.3E-10     PBSWG
= 0.8002027\n');
fprintf(fid,'+MJSWG   = 0.6         CF      = 0          PVTH0
= -7.55492E-4\n');
fprintf(fid,'+PRDSW   = 0          PK2     = 1.510875E-3  WKETA
= 0.0128032\n');
fprintf(fid,'+LKETA   = 0.0141455   PU0     = -3.9867632   PUA
= -3.10294E-11\n');
fprintf(fid,'+PUB     = 0          PVSAT   = 653.2294237 PETA0
= 1E-4\n');
fprintf(fid,'+PKETA   = -9.928283E-3 SAREF   = 5.5E-7       SBREF
= 5.5E-7\n');
fprintf(fid,'+WLOD    = 2E-6         KU0     = -4E-6        KVSAT
= 0.2\n');
fprintf(fid,'+KVTH0   = 2E-8         LLODKU0 = 1.0867072     STIMOD
= 2\n');
fprintf(fid,'+WLODKU0 = 1.0990864   LLODVTH = 1
WLODVTH = 1\n');
fprintf(fid,'+LKU0    = 1E-6         WKU0    = 1E-6
LODETA0 = 1\n');
fprintf(fid,'+LKVTH0  = 1.1E-6      WKVTH0  = 1.1E-6     PKVTH0
= 0\n');
fprintf(fid,'+STK2    = 0          LODK2   = 1          STETA0
= 0\n\n');

```

```

fprintf(fid, '.MODEL P PMOS (                                LEVEL =
8)\n');
fprintf(fid, '+VERSION = 3.1                                TNOM = 27                                TOX
= 3.2E-9\n');
fprintf(fid, '+XJ = 1E-7                                NCH = 4.1589E17                                VTH0
= -0.2176929\n');
fprintf(fid, '+K1 = 0.2566012                                K2 = 6.227669E-3                                K3
= 9.8203065\n');
fprintf(fid, '+K3B = 20                                W0 = 1E-6                                NLX
= 2.875116E-7\n');
fprintf(fid, '+DVTOW = 0                                DVT1W = 0                                DVT2W
= 0\n');
fprintf(fid, '+DVT0 = 0.1725649                                DVT1 = 0.2214064                                DVT2
= 0.1\n');
fprintf(fid, '+U0 = 105.8627093                                UA = 1.102916E-9                                UB
= 3.61005E-21\n');
fprintf(fid, '+UC = -4.6651E-11                                VSAT = 7.368516E4                                A0
= 2\n');
fprintf(fid, '+AGS = 0.7106176                                B0 = 4.50136E-6                                B1
= 5E-6\n');
fprintf(fid, '+KETA = 0.029826                                A1 = 0.1616816                                A2
= 0.3\n');
fprintf(fid, '+RDSW = 106.225597                                PRWG = -0.4457476                                PRWB
= 0.5\n');
fprintf(fid, '+WR = 1                                WINT = 0                                LINT
= 1.483182E-8\n');
fprintf(fid, '+DWG = 9.329926E-10                                DWB = -3.202354E-8                                VOFF
= -0.1022829\n');
fprintf(fid, '+NFACTOR = 1.5332272                                CIT = 0                                CDSC
= 2.4E-4\n');
fprintf(fid, '+CDSCD = 0                                CDSCB = 0                                ETA0
= 0.0115131\n');
fprintf(fid, '+ETAB = -6.181367E-3                                DSUB = 3.781207E-3                                PCLM
= 0.8846898\n');
fprintf(fid, '+PDIBLC1 = 8.605446E-4                                PDIBLC2 = 2.244662E-13
PDIBLCB = 0.1\n');
fprintf(fid, '+DROUT = 0                                PSCBE1 = 3.388133E9                                PSCBE2
= 1.039776E-9\n');
fprintf(fid, '+PVAG = 0                                DELTA = 0.01                                RSH
= 6.4\n');
fprintf(fid, '+MOBMOD = 1                                PRT = 0                                UTE
= -1.5\n');
fprintf(fid, '+KT1 = -0.11                                KT1L = 0                                KT2
= 0.022\n');
fprintf(fid, '+UA1 = 4.31E-9                                UB1 = -7.61E-18                                UC1
= -5.6E-11\n');
fprintf(fid, '+AT = 3.3E4                                WL = 0                                WLN
= 1\n');
fprintf(fid, '+WW = 0                                WWN = 1                                WWL
= 0\n');
fprintf(fid, '+LL = 0                                LLN = 1                                LW
= 0\n');
fprintf(fid, '+LWN = 1                                LWL = 0                                CAPMOD
= 2\n');
fprintf(fid, '+XPART = 0.5                                CGDO = 3.42E-10                                CGSO
= 3.42E-10\n');

```

```

fprintf(fid, '+CGBO      = 1E-12          CJ      = 1.155922E-3    PB
= 0.8\n');
fprintf(fid, '+MJ       = 0.4423239      CJSW     = 1.107787E-10    PBSW
= 0.8\n');
fprintf(fid, '+MJSW     = 0.1165941      CJSWG    = 4.22E-10         PBSWG
= 0.8\n');
fprintf(fid, '+MJSWG    = 0.1165941      CF       = 0              PVTH0
= 4.037191E-4\n');
fprintf(fid, '+PRDSW    = 58.2542288      PK2      = 2.351412E-3     WKETA
= 0.0350269\n');
fprintf(fid, '+LKETA    = 0.0229421      PU0      = -0.8           PUA
= -4.64248E-11\n');
fprintf(fid, '+PUB      = 0              PVSAT    = 50           PETA0
= 1E-4\n');
fprintf(fid, '+PKETA    = -0.0142482          )\n\n');
fprintf(fid, '.OP EPS_V=1u VMIN=0 VMAX=1.5 DELTAV=150m EPS_I=100p
MAXITER=500 HEURISTICS=-4 TRYALL=YES TIMEOUT=-1\n');
fprintf(fid, '.EPS 1u 100m 1n\n');
fprintf(fid, '.H 1fs 1fs 100ns 125m 2\n');
fprintf(fid, '.METHOD GEAR\n');
textocir=0;
type ACM.cir;
fclose(fid); % fecha e grava o arquivo para o SMASH
end % função acmsmash

% Esta função verifica os resultados da otimização calculados para
cada restrição
function
[verifica]=testa(x1,x2,x3,x4,x5,Vx,phit,IsqN,IsqP,VtoN,VtoP,nN,nP,Vss,
Vdd,lb,ub)
x(1)=x1;x(2)=x2;x(3)=x3;x(4)=x4;x(5)=x5;
display('Verificação dos resultados finais:');
display('Verifica restrições de igualdades não lineares ceq(1) e
ceq(2)');
%Verifica ceq(1)
ladosq=(Vx-Vdd-VtoP)/nP;
ladodir=-phit*(sqrt(1+x(5)*x(1)/(IsqP*x(3)))-
3+log(sqrt(1+x(5)*x(1)/(IsqP*x(3)))-1));
fprintf('ceq(1):\n(Vx-Vdd-VtoP)/nP=
%10.16f\n=\n-phit*(sqrt(1+x(5)*x(1)/(IsqP*x(3)))-
3+log(sqrt(1+x(5)*x(1)/(IsqP*x(3)))-1))1= %10.16f\n\n',ladosq,
ladodir);
% Verifica ceq(2)
ladosq=(Vx-Vss-VtoN)/nN;
ladodir=phit*(sqrt(1+x(5)*x(2)/(IsqN*x(4)))-
3+log(sqrt(1+x(5)*x(2)/(IsqN*x(4)))-1));
fprintf('ceq(2):\n(Vx-Vss-VtoN)/nN=
%10.16f\n=\nphit*(sqrt(1+x(5)*x(2)/(IsqN*x(4)))-
3+log(sqrt(1+x(5)*x(2)/(IsqN*x(4)))-1))= %10.16f\n\n',ladosq,
ladodir);
display('-----');
display('-----');
display('Verifica os limites (bounds) das variáveis:');
fprintf('%10.16f <=L1\n%10.16f = L1\n%10.16f
>=L1\n\n',lb(1),x(1),ub(1));

```

```
fprintf('%10.16f <=L2\n%10.16f = L2\n%10.16f
>=L2\n\n',lb(2),x(2),ub(2));
fprintf('%10.16f <=L3\n%10.16f = W1\n%10.16f
>=L3\n\n',lb(3),x(3),ub(3));
fprintf('%10.16f <=L4\n%10.16f = W2\n%10.16f
>=L4\n\n',lb(4),x(4),ub(4));
fprintf('%10.16f <= L5\n%10.16f = Id\n%10.16f >=
L5\n\n',lb(5),x(5),ub(5));
%display('O arquivo acm.cir foi gravado com os dados para entrada no
programa SMASH v5');
%display(' ');
%display('Grato por usar este programa. ');
%display('Ricardo Silva');
display('=====');
display('=====');
verifica=0;
display(' ');
end % função testa
```

## B.2 Projeto do Integrador de Seevinck

```

% Minimização do Circuito Seevinck
function [dummy]=seevarea(~)
clc;
clear all
display('OTIMIZAÇÃO DE CIRCUITOS INTEGRADOS - MATLAB OPTIMIZATION
TOOLBOX - KNITRO');
display('MÉTODOS DOS PONTOS INTERIORES');
% Valores sugeridos para entrada pelo usuário: Kint, iinmax e iinmin.
% Kint: constante de integração, numerador da função de transferência
T(s)= Kint/s (radianos por segundo)
% iinmax e iinmin (Amperes)
% Exemplo A
% Kint=2*pi*10^5
% iinmax=2e-7
% iinmin=-2e-7
% Exemplo B
% Kint=2*pi*10^5
% iinmax=2e-8
% iinmin=-2e-8
% Exemplo C
% Kint=2*pi*10^6
% iinmax=2e-8
% iinmin=-2e-8

Kint=input('Digite o valor de "Kint" em radianos por segundo: \n? ');
if Kint <=0
    display('Erro');return
end
iinmax=input('Digite o valor de "iinmax" em Amperes: \n? ');
if iinmax <=0
    display('Erro');return
end
iinmin=input('Digite o valor de "iinmin" em Amperes: \n? ');
NIter=input('Defina a quantidade máxima de iterações do KNITRO: \n?
');
if NIter <=0
    display('Erro');return
end
% Tolerâncias
tolx=1e-15; % tolerância para as variáveis
tolfun=1e-06; % optimality error (função objetivo)
tolcon=1e-06; % tolerância para a região factível das restrições
fprintf('\nOs valores "default" das tolerâncias são:\nTolX= 1e-15
(variáveis)\nTolFun=1e-06 (função objetivo)\nTolCon=1e-06 (região
factível das restrições)');
fprintf('\n\nEscolha se quer mudar as tolerâncias.\n');
rel=input('Digite "S" ou "N": ','s');
if strcmp(rel,'S')==1
    tolx=input('Defina a tolerância das variáveis (TolX): \n? ');
    if tolx <=0 display('Erro');return
end
end

```

```

    tolfun=input('Defina a tolerância da função objetivo (TolFun): \n?
');
    if tolfun <=0 display('Erro');return
    end
    tolcon=input('Defina a tolerância das restrições (TolCon): \n? ');
    if tolcon <=0
        display('Erro');return
    end
elseif strcmp(rel,'N')==1
else display('Erro');return
end
display(' ');
display('=====');
fprintf('Cálculo da Otimização com o MATLAB Optimization Toolbox/KNITRO
\n');
disp('Método da Otimização: Pontos Interiores');

% Valores iniciais
% I(A) Io(A) Ibias(A) L(m) W1=W2(m) W3(m)
W4(m) W6=W11(m)W7=W8(m) W10(m) C(F)
x0=[226*10^-9; 226*10^-9; 500*10^-9; 0.5*10^-6; 10*10^-6; 10*10^-6;
10*10^-6; 5*10^-6; 5*10^-6; 10*10^-6; 20*10^-12];

% Constantes (transistores canal N da tecnologia 0.13 micrômetros IBM)
aa=1.0382;
bb=4.4688;
phit=0.26e-1;
Co=2.1128*10^(-3); % Farads/metro quadrado
Isq=353.38*10^(-9); % Amperes
Vto=.3618; % Volts
n=1.3790;

% Passa as restrições "limites ("bounds") das variáveis" para o
formato do toolbox
% Limites inferiores
% I(A) Io(A) Ibias(A) L(m) W1=W2(m) W3(m)
W4(m) W6=W11(m) W7=W8(m) W10(m) C(F)
lb=[1*10^(-12);1*10^(-12);1*10^(-12);0.26*10^(-6);0.26*10^(-
6);0.26*10^(-6);0.26*10^(-6);0.26*10^(-6);0.26*10^(-
6);10*10^(-12)];
% Limites superiores
% I(A) Io(A) Ibias(A) L(m) W1=W2(m) W3(m)
W4(m) W6=W11(m) W7=W8(m) W10(m) C(F)
ub=[10*10^(-6);10*10^(-6);10*10^(-6);100*10^(-6);100*10^(-6);100*10^(-
6);100*10^(-6);100*10^(-6);100*10^(-6);100*10^(-6);100*10^(-12)];
% Igualdades lineares (coeficientes)
Aeq=[];
beq=[];
% Desigualdades lineares (coeficientes) ("<=")
A=[0,0,-1,0,0,0,0,0,0,0,0];
b=[iinmin/2];

% Mostra os dados de entrada fornecidos
display(' ');
display('Dados de entrada:');

```

```

display(['Kint=',num2str(Kint),' radianos por segundo']);
display(['iinmax=',num2str(iinmax),' Amperes']);
display(['iinmin=',num2str(iinmin),' Amperes']);
display(' ');
fprintf('Valores iniciais das variáveis:\nI=%3.5s Io=%3.5s
Ibias=%3.5s \nL=%1.5s W1=W2=%2.5s W3=%2.5s \nW4=%2.5s W6=W11=%1.5s
W7=W8=%1.5s \nW10=%2.5s
C=%2.5s\n',x0(1),x0(2),x0(3),x0(4),x0(5),x0(6),x0(7),x0(8),x0(9),x0(10)
),x0(11));
fprintf('\nLimites ("bounds") inferiores e superiores:\n(Unidades:
Amperes, metros e Farads)\n%1.3s <= I <= %2.3s\n%1.3s <= Io
<= %2.3s\n%1.3s <= Ibias <= %2.3s\n%1.3s <= L <= %3.3s\n%1.3s <=
W1=W2 <= %3.3s\n%1.3s <= W3 <= %3.3s\n%1.3s <= W4 <=
%3.3s\n%1.3s <= W6=W11 <= %3.3s\n%1.3s <= W7=W8 <= %3.3s\n%1.3s <=
W10 <= %3.3s\n%2.3s <= C <=
%3.3s\n',lb(1),ub(1),lb(2),ub(2),lb(3),ub(3),lb(4),ub(4),lb(5),ub(5),l
b(6),ub(6),lb(7),ub(7),lb(8),ub(8),lb(9),ub(9),lb(10),ub(10),lb(11),ub
(11));
fprintf('\nTolerâncias especificadas:\nTolX= %1.0e
(variáveis)\nTolFun=%1.0e (função objetivo)\nTolCon=%1.0e (região
factível das restrições)\n',tolx,tolfun,tolcon);
display(' ');
display('=====');
display('Resultado das iterações:');
% Calcula iterações e gera relatório detalhado
[x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,lb,ub,aa,bb,phit,Kint,iinmin,iinma
x,Isq,Vto,n]=iterotima(x0,Kint,iinmax,iinmin,aa,bb,phit,Co,Isq,Vto,n,l
b,ub,Aeq,beq,A,b,NIter,tolx,tolfun,tolcon);
% Grava arquivo para a simulação do circuito no SMASH
[textocir]=seevsmash(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11);
% Verificação dos resultados finais
[nada]=testa(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,lb,ub,aa,bb,phit,Kint,
iinmin,iinmax,Isq,Vto,n);
end % FIM DO PROGRAMA

function
[x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,lb,ub,aa,bb,phit,Kint,iinmin,iinma
x,Isq,Vto,n]=iterotima(x0,Kint,iinmax,iinmin,aa,bb,phit,Co,Isq,Vto,n,l
b,ub,Aeq,beq,A,b,NIter,tolx,tolfun,tolcon)
% Esta função calcula a otimização e gera um relatório das iterações
% Opções do Toolbox "knitro"
options =
optimset('AlwaysHonorConstraints','bounds','Algorithm','interior-
point','InitBarrierParam',0.1,'Display','iter','GradObj','on','GradCon
str','on','Hessian','bfgs','MaxIter',NIter,'TolX',tolx,'TolFun',tolfun
,'TolCon',tolcon);
% Chama KNITRO para resolver o modelo de otimização.
[x,fval]=ktrlink(@objfun,x0,A,b,Aeq,beq,lb,ub,@constfun,options);
Area=2*x(4)*(2*x(5)+x(6)+x(7)+2*x(9)+x(10)+2*x(8))+2*x(11)/Co;
x1=x(1);x2=x(2);x3=x(3);x4=x(4);x5=x(5);x6=x(6);x7=x(7);x8=x(8);x9=x(9)
);x10=x(10);x11=x(11);

% Função objetivo
function [f,g] = objfun(x)
f=2*x(4)*(2*x(5)+x(6)+x(7)+2*x(9)+x(10)+2*x(8))+2*x(11)/Co;

```

```

%Gradiente da função objetivo
g=[0;0;0;4*x(5)+2*x(6)+2*x(7)+4*x(9)+2*x(10)+4*x(8);4*x(4);2*x(4);2*x(
4);4*x(4);4*x(4);2*x(4);2/Co];
end %função objfun
function [c,ceq,Gc,Gceq]= constfun(x)
% Funções de restrições não-lineares (Desigualdades '<=0')
c=[(x(1)/x(2))-(x(6)*exp(4/aa)/x(10))+1
-(((x(3)+(1/2)*iinmin)*x(4)*((x(1)+x(2))*x(10))/(x(2)*x(6))-
1))/x(5))+((x(1)+x(2))*x(4)*x(10))/(x(5)*x(6))+bb*Isq*exp((4*phit-
Vto)/(n*aa*phit))
-x(5)/x(4)+(1.667*((1-
x(6)*x(2)/(x(10)*(x(1)+x(2))))*(x(3)+(1/2)*iinmax)-x(2)))/Isq
-x(6)/x(4)+(1.667*(x(1)+x(2)))/Isq
-x(9)/x(4)+(1.667*(x(3)+(1/2)*iinmax))/Isq
-x(10)/x(4)+1.667*x(2)/Isq
-x(8)/x(4)+1.667*x(2)/Isq];
% Funções de restrições não-lineares (Igualdades '=0')
ceq=[x(6)/x(10)-x(7)/x(8)
((x(1)+x(2)-x(6)*x(2)/x(10))/(x(11)*aa*phit))-Kint];
% Gradiente das desigualdades não-lineares 'c'
Gc1=[1/x(2)
-x(1)/(x(2)^2)
0
0
0
-exp(4/aa)/x(10)
0
0
0
x(6)*exp(4/aa)/(x(10)^2)
0];
Gc2=[-
((x(3)+(1/2)*iinmin)*x(4)*x(10))/(x(5)*x(2)*x(6))+((x(4)*x(10))/(x(5)*x
(6))
-(((x(3)+(1/2)*iinmin)*x(4)*((x(10))/(x(2)*x(6))-
((x(1)+x(2))*x(10))/((x(2)^2)*x(6))))/(x(5))+((x(4)*x(10))/(x(5)*x(6))
-x(4)*(((x(1)+x(2))*x(10))/(x(2)*x(6))-1))/x(5)
-(((x(3)+(1/2)*iinmin)*((x(1)+x(2))*x(10))/(x(2)*x(6))-
1))/x(5))+((x(1)+x(2))*x(10))/(x(5)*x(6))
((x(3)+(1/2)*iinmin)*x(4)*((x(1)+x(2))*x(10))/(x(2)*x(6))-
1))/((x(5)^2)-((x(1)+x(2))*x(4)*x(10))/((x(5)^2)*x(6))
((x(3)+(1/2)*iinmin)*x(4)*(x(1)+x(2))*x(10))/(x(5)*x(2)*(x(6)^2))-
((x(1)+x(2))*x(4)*x(10))/(x(5)*(x(6)^2))
0
0
0
-
((x(3)+(1/2)*iinmin)*x(4)*(x(1)+x(2)))/(x(5)*x(2)*x(6))+((x(1)+x(2))*x
(4))/(x(5)*x(6))
0];
Gc3=[1.667*x(6)*x(2)*(x(3)+(1/2)*iinmax)/(x(10)*((x(1)+x(2))^2)*Isq)
(1.667*((-
x(6)/((x(1)+x(2))*x(10))+x(6)*x(2)/(x(10)*(x(1)+x(2))^2))*(x(3)+(1/2)*
iinmax)-1))/Isq
(1.667-1.667*x(6)*x(2)/(x(10)*(x(1)+x(2))))/Isq

```

```

x(5)/(x(4)^2)
-1/x(4)
-1.667*x(2)*(x(3)+(1/2)*iinmax)/(x(10)*(x(1)+x(2))*Isq)
0
0
0
1.667*x(6)*x(2)*(x(3)+(1/2)*iinmax)/((x(10)^2)*(x(1)+x(2))*Isq)
0];
Gc4=[1.667/Isq
1.667/Isq
0
x(6)/(x(4)^2)
0
-1/x(4)
0
0
0
0
0];
Gc5=[0
0
1.667/Isq
x(9)/(x(4)^2)
0
0
0
0
-1/x(4)
0
0];
Gc6=[0
1.667/Isq
0
x(10)/(x(4)^2)
0
0
0
0
-1/x(4)
0];
Gc7=[0
1.667/Isq
0
x(8)/(x(4)^2)
0
0
0
-1/x(4)
0
0
0];
Gc=[Gc1, Gc2, Gc3, Gc4, Gc5, Gc6, Gc7];
% Gradiente das igualdades não-lineares 'ceq'
Gceq1=[0
0

```

```

0
0
0
1/x(10)
-1/x(8)
x(7)/(x(8)^2)
0
-x(6)/(x(10)^2)
0];
Gceq2=[1/(x(11)*aa*phit)
(1-x(6)/x(10))/(x(11)*aa*phit)
0
0
0
-x(2)/x(11)*aa*phit*x(10)
0
0
0
x(6)*x(2)/(x(11)*aa*phit*(x(10)^2))
-(x(1)+x(2)-x(6)*x(2)/x(10))/((x(11)^2)*aa*phit)];
Gceq=[Gceq1, Gceq2];
end % função constfun
end % função iterotima

function [textocir]=seevsmash(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11)
% Esta função grava um arquivo do circuito otimizado para ser usado no
Smash
Vdd=0.6; % Volts
Vss=-0.6; % Volts
textocir='seev.cir'; % nome do arquivo de descrição do circuito
fid = fopen(textocir,'w'); % cria arquivo-texto para o SMASH
fprintf(fid,'* Arquivo de simulação no SMASH (baseado em ideia de
Victor Ariel-UFBA)\n');
fprintf(fid,'* Unidades: metro, Volt, Ampere, Farad, Ohm \n');
fprintf(fid,'* Integrador de Seevinck \n');
fprintf(fid,'* %s \r\n',datestr(now,'dd/mm/yyyy HH:MM:SS \n'));
fprintf(fid,'M1A DD G1A S1A SS N W=%e L=%e \n',x5,x4);
fprintf(fid,'M2A DD G1A S2A SS N W=%e L=%e \n',x5,x4);
fprintf(fid,'M3A G1A G3A SS SS N W=%e L=%e \n',x6,x4);
fprintf(fid,'M4A D4A G4A SS SS N W=%e L=%e \n',x7,x4);
fprintf(fid,'M6A S2A G4A SS SS N W=%e L=%e \n',x8,x4);
fprintf(fid,'M7A D7A D7A SS SS N W=%e L=%e \n',x9,x4);
fprintf(fid,'M8A S1A D7A SS SS N W=%e L=%e \n',x9,x4);
fprintf(fid,'M10A G3A G3A S1A SS N W=%e L=%e \n',x10,x4);
fprintf(fid,'M11A G4A G4A S2B SS N W=%e L=%e \n',x8,x4);
fprintf(fid,'CINTA S2B SS %e \n \n',x11);
fprintf(fid,'M1B DD G1B S1B SS N W=%e L=%e \n',x5,x4);
fprintf(fid,'M2B DD G1B S2B SS N W=%e L=%e \n',x5,x4);
fprintf(fid,'M3B G1B G3B SS SS N W=%e L=%e \n',x6,x4);
fprintf(fid,'M4B D4B G4B SS SS N W=%e L=%e \n',x7,x4);
fprintf(fid,'M6B S2B G4B SS SS N W=%e L=%e \n',x8,x4);
fprintf(fid,'M7B D7B D7B SS SS N W=%e L=%e \n',x9,x4);
fprintf(fid,'M8B S1B D7B SS SS N W=%e L=%e \n',x9,x4);
fprintf(fid,'M10B G3B G3B S1B SS N W=%e L=%e \n',x10,x4);
fprintf(fid,'M11B G4B G4B S2A SS N W=%e L=%e \n',x8,x4);
fprintf(fid,'CINTB S2A SS %e \n \n',x11);

```

```

fprintf(fid,'VDD DD 0 DC %eV \n',Vdd);
fprintf(fid,'VSS SS 0 DC %eV \n\n',Vss);
fprintf(fid,'.PARAM frac=1 \n');
fprintf(fid,'II0A DD G1A DC %e \n',x1+x2);
fprintf(fid,'I01A DD G3A DC %e \n',x2);
fprintf(fid,'I02A DD G4A DC %e \n',x2);
fprintf(fid,'IBIASA DD D7A %e \n',x3);
fprintf(fid,'IINA DD D7A AC 1 0 \n');
fprintf(fid,'VOUTA D4A 0 DC 0 \n\n');
fprintf(fid,'II0B DD G1B DC %e \n',x1+x2);
fprintf(fid,'I01B DD G3B DC %e \n',x2);
fprintf(fid,'I02B DD G4B DC %e \n',x2);
fprintf(fid,'IBIASB DD D7B %e \n',x3);
fprintf(fid,'IINB D7B DD AC 1 0 \n');
fprintf(fid,'VOUTB D4B 0 DC 0 \n\n');
fprintf(fid,'.AC DEC 100 100 100MEG \n'); % 100 pontos por década /
entre 100 Hz e 100 MHz
fprintf(fid,'.TRAN 100ns 2000us 1000us \n');
fprintf(fid,'.Trace AC {IOUTDB = DB((I(VOUTB) - I(VOUTA))/2)}
\n');
fprintf(fid,'.Trace TRAN {IOUT = I(VOUTB) - I(VOUTA)} \n');
fprintf(fid,'.Trace TRAN I(IINA1) I(IINA2) \n');
fprintf(fid,'.CAPAMIN 1e-18 fF \n\n');
fprintf(fid,'* Dados da Tecnologia \r\n');
fprintf(fid,'* BSIM3v3.1 parameters for IBM 0.13 \n');
fprintf(fid,'* Technology: SCN013 \n');
fprintf(fid,'*RUN: T93V (#2_8RF_8LM_DM)\n');
fprintf(fid,'*SPICE 3f5 Level 8, Star-HSPICE Level 49, UTMOST Level
8\n');
fprintf(fid,'.MODEL N NMOS ( LEVEL =
8\n');
fprintf(fid,'+VERSION = 3.1 TNOM = 27 TOX
= 3.2E-9\n');
fprintf(fid,'+XJ = 1E-7 NCH = 2.3549E17 VTH0
= 0.0423138\n');
fprintf(fid,'+K1 = 0.3322485 K2 = -0.0298343 K3
= 1E-3\n');
fprintf(fid,'+K3B = 3.6987372 W0 = 1E-7 NLX
= 1E-6\n');
fprintf(fid,'+DVT0W = 0 DVT1W = 0 DVT2W
= 0\n');
fprintf(fid,'+DVT0 = 1.2848863 DVT1 = 0.1532141 DVT2
= 0.2740377\n');
fprintf(fid,'+U0 = 446.4279855 UA = -1.40947E-10 UB
= 3.152185E-18\n');
fprintf(fid,'+UC = 4.34007E-10 VSAT = 7.64383E4 A0
= 0.1112707\n');
fprintf(fid,'+AGS = 0.0734135 B0 = 3.913953E-6 B1
= 5E-6\n');
fprintf(fid,'+KETA = 0.05 A1 = 1.042383E-3 A2
= 0.7518921\n');
fprintf(fid,'+RDSW = 150 PRWG = 0.3496099 PRWB
= 0.1112769\n');
fprintf(fid,'+WR = 1 WINT = 2.378741E-10 LINT
= 1.040187E-8\n');

```

```

fprintf(fid,'+DWG      = 1.464878E-8      DWB      = 9.306917E-9      VOFF
= -0.0240657\n');
fprintf(fid,'+NFACTOR = 2.5              CIT      = 0              CDSC
= 2.4E-4\n');
fprintf(fid,'+CDSCD   = 0              CDSCB   = 0              ETA0
= 2.750169E-6\n');
fprintf(fid,'+ETAB    = -0.0151402      DSUB    = 4.056696E-6      PCLM
= 1.9784063\n');
fprintf(fid,'+PDIBLC1 = 0.9657497      PDIBLC2 = 0.01
PDIBLCB = 0.1\n');
fprintf(fid,'+DROUT   = 0.9990298      PSCBE1  = 7.957109E10     PSCBE2
= 5.015981E-10\n');
fprintf(fid,'+PVAG    = 0.535376       DELTA   = 0.01          RSH
= 6.4\n');
fprintf(fid,'+MOBMOD  = 1              PRT     = 0              UTE
= -1.5\n');
fprintf(fid,'+KT1     = -0.11          KT1L    = 0              KT2
= 0.022\n');
fprintf(fid,'+UA1     = 4.31E-9        UB1     = -7.61E-18       UC1
= -5.6E-11\n');
fprintf(fid,'+AT      = 3.3E4         WL      = 0              WLN
= 1\n');
fprintf(fid,'+WW      = 0              WWN     = 1              WWL
= 0\n');
fprintf(fid,'+LL      = 0              LLN     = 1              LW
= 0\n');
fprintf(fid,'+LWN     = 1              LWL     = 0              CAPMOD
= 2\n');
fprintf(fid,'+XPART   = 0.5           CGDO    = 3.74E-10         CGSO
= 3.74E-10\n');
fprintf(fid,'+CGBO    = 1E-12         CJ      = 9.581155E-4     PB
= 0.9756037\n');
fprintf(fid,'+MJ      = 0.4044076     CJSW    = 1E-10         PBSW
= 0.8002027\n');
fprintf(fid,'+MJSW    = 0.6           CJSWG   = 3.3E-10         PBSWG
= 0.8002027\n');
fprintf(fid,'+MJSWG   = 0.6           CF      = 0              PVTH0
= -7.55492E-4\n');
fprintf(fid,'+PRDSW   = 0           PK2     = 1.510875E-3     WKETA
= 0.0128032\n');
fprintf(fid,'+LKETA   = 0.0141455     PU0     = -3.9867632      PUA
= -3.10294E-11\n');
fprintf(fid,'+PUB     = 0           PVSAT   = 653.2294237    PETA0
= 1E-4\n');
fprintf(fid,'+PKETA   = -9.928283E-3   SAREF   = 5.5E-7         SBREF
= 5.5E-7\n');
fprintf(fid,'+WLOD    = 2E-6           KU0     = -4E-6          KVSAT
= 0.2\n');
fprintf(fid,'+KVTH0   = 2E-8           LLODKU0 = 1.0867072        STIMOD
= 2\n');
fprintf(fid,'+WLODKU0 = 1.0990864     LLODVTH = 1
WLODVTH = 1\n');
fprintf(fid,'+LKU0    = 1E-6           WKU0    = 1E-6
LODETA0 = 1\n');
fprintf(fid,'+LKVTH0 = 1.1E-6       WKVTH0  = 1.1E-6       PKVTH0
= 0\n');

```

```

fprintf(fid,'+STK2      = 0          LODK2      = 1          STETA0
= 0\n\n');
fprintf(fid,'.MODEL P PMOS (          LEVEL      =
8)\n');
fprintf(fid,'+VERSION = 3.1          TNOM       = 27          TOX
= 3.2E-9\n');
fprintf(fid,'+XJ       = 1E-7          NCH       = 4.1589E17      VTH0
= -0.2176929\n');
fprintf(fid,'+K1       = 0.2566012     K2        = 6.227669E-3     K3
= 9.8203065\n');
fprintf(fid,'+K3B      = 20           W0        = 1E-6          NLX
= 2.875116E-7\n');
fprintf(fid,'+DVT0W    = 0           DVT1W     = 0          DVT2W
= 0\n');
fprintf(fid,'+DVT0     = 0.1725649     DVT1      = 0.2214064     DVT2
= 0.1\n');
fprintf(fid,'+U0       = 105.8627093    UA        = 1.102916E-9      UB
= 3.61005E-21\n');
fprintf(fid,'+UC       = -4.6651E-11    VSAT     = 7.368516E4      A0
= 2\n');
fprintf(fid,'+AGS      = 0.7106176     B0        = 4.50136E-6      B1
= 5E-6\n');
fprintf(fid,'+KETA     = 0.029826      A1        = 0.1616816      A2
= 0.3\n');
fprintf(fid,'+RDSW     = 106.225597    PRWG     = -0.4457476      PRWB
= 0.5\n');
fprintf(fid,'+WR       = 1           WINT     = 0          LINT
= 1.483182E-8\n');
fprintf(fid,'+DWG      = 9.329926E-10   DWB      = -3.202354E-8    VOFF
= -0.1022829\n');
fprintf(fid,'+NFACTOR  = 1.5332272     CIT       = 0          CDSC
= 2.4E-4\n');
fprintf(fid,'+CDSCD    = 0           CDSCB    = 0          ETA0
= 0.0115131\n');
fprintf(fid,'+ETAB     = -6.181367E-3    DSUB     = 3.781207E-3    PCLM
= 0.8846898\n');
fprintf(fid,'+PDIBLC1  = 8.605446E-4     PDIBLC2  = 2.244662E-13
PDIBLCB = 0.1\n');
fprintf(fid,'+DROUT    = 0           PSCBE1   = 3.388133E9      PSCBE2
= 1.039776E-9\n');
fprintf(fid,'+PVAG     = 0           DELTA    = 0.01         RSH
= 6.4\n');
fprintf(fid,'+MOBMOD   = 1           PRT      = 0          UTE
= -1.5\n');
fprintf(fid,'+KT1      = -0.11         KT1L     = 0          KT2
= 0.022\n');
fprintf(fid,'+UA1      = 4.31E-9        UB1      = -7.61E-18      UC1
= -5.6E-11\n');
fprintf(fid,'+AT       = 3.3E4         WL       = 0          WLN
= 1\n');
fprintf(fid,'+WW       = 0           WWN      = 1          WWL
= 0\n');
fprintf(fid,'+LL       = 0           LLN      = 1          LW
= 0\n');
fprintf(fid,'+LWN      = 1           LWL      = 0          CAPMOD
= 2\n');

```

```

fprintf(fid,'+XPART    = 0.5                CGDO    = 3.42E-10        CGSO
= 3.42E-10\n');
fprintf(fid,'+CGBO    = 1E-12              CJ       = 1.155922E-3       PB
= 0.8\n');
fprintf(fid,'+MJ      = 0.4423239          CJSW    = 1.107787E-10    PBSW
= 0.8\n');
fprintf(fid,'+MJSW    = 0.1165941          CJSWG   = 4.22E-10          PBSWG
= 0.8\n');
fprintf(fid,'+MJSWG   = 0.1165941          CF      = 0                PVTH0
= 4.037191E-4\n');
fprintf(fid,'+PRDSW   = 58.2542288         PK2     = 2.351412E-3       WKETA
= 0.0350269\n');
fprintf(fid,'+LKETA   = 0.0229421          PU0     = -0.8              PUA
= -4.64248E-11\n');
fprintf(fid,'+PUB     = 0                  PVSAT   = 50            PETA0
= 1E-4\n');
fprintf(fid,'+PKETA   = -0.0142482          )\n\n');
fprintf(fid,'.OP EPS_V=1u VMIN=0 VMAX=1.5 DELTAV=150m EPS_I=100p
MAXITER=500 HEURISTICS=-4 TRYALL=YES TIMEOUT=-1\n');
fprintf(fid,'.EPS 1u 100m 1n\n');
fprintf(fid,'.H 1fs 1fs 100ns 125m 2\n');
fprintf(fid,'.METHOD GEAR\n');
type seev.cir;
fclose(fid); % fecha e grava o arquivo para o SMASH
end % função seevsmash

```

```

function[nada]=testa(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,lb,ub,aa,bb,ph
it,Kint,iinmin,iinmax,Isq,Vto,n)
% Esta função verifica os resultados da otimização calculados para
cada restrição
x(1)=x1;x(2)=x2;x(3)=x3;x(4)=x4;x(5)=x5;x(6)=x6;x(7)=x7;x(8)=x8;x(9)=x
9;x(10)=x10;x(11)=x11;
display('-----');
display('Verificação dos resultados finais:');
display('Verifica restrições de igualdades não-lineares ceq(1) e
ceq(2)');
%Verifica ceq(1)
ladosq=(x(6)/x(10));
ladodir=(x(7)/x(8));
fprintf('ceq(1):\nW3/W10= %10.16f\n=\nW4/W11= %10.16f\n\n',ladosq,
ladodir);
% Verifica ceq(2)
ladosq=((x(1)+x(2)-x(6)*x(2)/x(10))/(x(11)*aa*phit));
ladodir=Kint;
fprintf('ceq(2):\n(1/C*a*phit)*(I+Io-W3/W10*Io)= %10.16f\n=\nKint=
%10.16f\n\n',ladosq, ladodir);
% Verifica a restrição de desigualdade linear Ibias>=-iinmin/2
display('Verificação da desigualdade linear');
ladosq=x(3);
ladodir=-iinmin/2;
fprintf('Desigualdade linear:\nIbias=          %10.16f\n>=\n-iinmin/2=
%10.16f\n\n',ladosq, ladodir);
% Verifica as restrições de desigualdades não-lineares c(1) a c(7)
display('Verificação das desigualdades não-lineares c(1) a c(7)');
% Verifica c(1)

```

```

ladosq=(x(1)/x(2));
ladodir=(x(6)*exp(4/aa)/x(10))-1;
fprintf('c(1):\nI/Io=           %10.16f\n<=\n(W3/W10)*exp(4/a)-
l= %10.16f\n\n',ladosq, ladodir);
% Verifica c(2)
ladosq=-(x(3)+(1/2)*iinmin)*x(4)*((x(1)+x(2))*x(10)/(x(2)*x(6))-
1)/x(5)+(x(1)+x(2))*x(4)*x(10)/(x(5)*x(6));
ladodir=bb*Isq*exp((4*phit-Vto)/(n*aa*phit));
fprintf('c(2):\n-
(x(3)+(1/2)*iinmin)*x(4)*((x(1)+x(2))*x(10)/(x(2)*x(6))-
1)/x(5)+(x(1)+x(2))*x(4)*x(10)/(x(5)*x(6))=\n
%10.16f\n',ladosq);
fprintf('>=\nbb*Isq*exp((4*phit-Vto)/(n*aa*phit))=
%10.16f\n\n',ladodir);
% Verifica c(3)
ladosq=x(5)/x(4);
ladodir=1.667*((1-x(6)*x(2)/(x(10)*(x(1)+x(2))))*(x(3)+(1/2)*iinmax)-
x(2))/Isq;
fprintf('c(3):\nW1/L=
%10.16f\n>=\n(1.667*((1-W3*Io/(W10*(I+Io)))*(Ibias+(1/2)*iinmax)-
Io))/Isq= %10.16f\n\n',ladosq, ladodir);
% Verifica c(4)
ladosq=x(6)/x(4);
ladodir=(1.667*(x(1)+x(2)))/Isq;
fprintf('c(4):\nW3/L=           %10.16f\n>=\n(1.667*(I+Io))/Isq=
%10.16f\n\n',ladosq, ladodir);
% Verifica c(5)
ladosq=x(9)/x(4);
ladodir=(1.667*(x(3)+(1/2)*iinmax))/Isq;
fprintf('c(5):\nW7/L=
%10.16f\n>=\n(1.667*(Ibias+(1/2)*iinmax))/Isq= %10.16f\n\n',ladosq,
ladodir);
% Verifica c(6)
ladosq=x(10)/x(4);
ladodir=1.667*x(2)/Isq;
fprintf('c(6):\nW10/L=           %10.16f\n>=\n1.667*Io/Isq=
%10.16f\n\n',ladosq, ladodir);
% Verifica c(7)
ladosq=x(8)/x(4);
ladodir=1.667*x(2)/Isq;
fprintf('c(7):\nW11/L=           %10.16f\n>=\n1.667*Io/Isq=
%10.16f\n\n',ladosq, ladodir);
display('Verifica os limites (bounds) das variáveis:');
fprintf('%10.16f <=I\n%10.16f = I\n%10.16f >=I\n\n',lb(1),x(1),ub(1));
fprintf('%10.16f <=Io\n%10.16f = Io\n%10.16f
>=Io\n\n',lb(2),x(2),ub(2));
fprintf('%10.16f <=Ibias\n%10.16f = Ibias\n%10.16f
>=Ibias\n\n',lb(3),x(3),ub(3));
fprintf('%10.16f <=L\n%10.16f = L\n%10.16f >=L\n\n',lb(4),x(4),ub(4));
fprintf('%10.16f <= W1=W2\n%10.16f = W1=W2\n%10.16f >=
W1=W2\n\n',lb(5),x(5),ub(5));
fprintf('%10.16f <= W3\n%10.16f = W3\n%10.16f >=
W3\n\n',lb(6),x(6),ub(6));
fprintf('%10.16f <= W4\n%10.16f = W4\n%10.16f >=
W4\n\n',lb(7),x(7),ub(7));

```

```
fprintf('%10.16f <=W6=W11\n%10.16f = W6=W11\n%10.16f
>=W6=W11\n\n',lb(8),x(8),ub(8));
fprintf('%10.16f <=W7=W8\n%10.16f = W7=W8\n%10.16f
>=W7=W8\n\n',lb(9),x(9),ub(9));
fprintf('%10.16f <= W10\n%10.16f = W10\n%10.16f >=
W10\n\n',lb(10),x(10),ub(10));
fprintf('%10.16f <=C\n%10.16f = C\n%10.16f
>=C\n\n',lb(11),x(11),ub(11));
display('O arquivo seev.cir foi gravado com os dados para entrada no
programa SMASH v5');
display(' ');
display('Grato por usar este programa.');
```

Ricardo Silva

```
nada=0;
display(' ');
end % função testa
```

**APÊNDICE C – Cálculos das iterações**

**C.1 Projeto do divisor de tensões**

OTIMIZAÇÃO DE CIRCUITOS INTEGRADOS - MATLAB OPTIMIZATION TOOLBOX –  
KNITRO - MÉTODOS DOS PONTOS INTERIORES

=====  
Digite o valor de "Vx" (em Volts):  
? .35  
Defina a quantidade máxima de iterações do KNITRO:  
? 50  
Os valores "default" das tolerâncias são:  
TolX= 1e-15 (variáveis)  
TolFun=1e-06 (função objetivo)  
TolCon=1e-06 (região factível das restrições)  
  
Escolha se quer mudar as tolerâncias.  
Digite "S" ou "N": N

=====  
Cálculo da Otimização com o MATLAB Opimization Toolbox/KNITRO  
Método da Otimização: Pontos Interiores

Limites ("bounds") inferiores e superiores:  
2.700e-007 <= L1 <= 1.000e-004 (metros)  
2.700e-007 <= L2 <= 1.000e-004 (metros)  
2.700e-007 <= W1 <= 1.000e-004 (metros)  
2.700e-007 <= W2 <= 1.000e-004 (metros)  
1.000e-012 <= Id <= 1.000e-004 (Amperes)

Tolerâncias especificadas:  
TolX= 1e-015 (variáveis)  
TolFun=1e-006 (função objetivo)  
TolCon=1e-006 (região factível das restrições)

=====  
EXEMPLO NÚMERO 1

Dado de entrada:  
Vx=0.35 Volts

Valores iniciais das variáveis:  
L1=2.70000e-007  
L2=2.70000e-007  
W1=2.70000e-007  
W2=2.70000e-007  
Id=1.00000e-012

=====  
Resultado das iterações:

KNITRO 7.0.0  
Ziena Optimization, Inc.

```

=====
algorithm:      1
hessopt:       2
honorbnds:     1
maxit:         50
outlev:        1
KNITRO changing bar_murule from AUTO to 4.
KNITRO changing bar_initpt from AUTO to 2.
KNITRO changing bar_penaltyrule from AUTO to 2.
KNITRO changing bar_penaltycons from AUTO to 1.
KNITRO changing linsolver from AUTO to 5.
KNITRO shifted start point to honor bounds (5 variables).

```

Problem Characteristics

```

-----
Objective goal: Minimize
Number of variables:      5
  bounded below:          0
  bounded above:          0
  bounded below and above: 5
  fixed:                  0
  free:                   0
Number of constraints:    2
  linear equalities:      0
  nonlinear equalities:   2
  linear inequalities:    0
  nonlinear inequalities: 0
  range:                  0
Number of nonzeros in Jacobian: 10
Number of nonzeros in Hessian: 15

```

EXIT: Locally optimal solution found.

Final Statistics

```

-----
Final objective value      = 4.82514530444625e-010
Final feasibility error (abs / rel) = 1.80e-015 / 1.80e-015
Final optimality error (abs / rel) = 1.17e-008 / 1.17e-008
# of iterations            = 16
# of CG iterations         = 1
# of function evaluations  = 22
# of gradient evaluations  = 17
Total program time (secs) = 0.248 ( 0.250 CPU time)
Time spent in evaluations (secs) = 0.182

```

```

=====
* Arquivo de simulação no SMASH (baseado em ideia de Victor Ariel-UFBA)
* Unidades: metro, Volt, Ampere, Farad, Ohm
* Divisor de Tensão - Modelo ACM
* 25/05/2011 20:44:23

```

```
M1 OUT OUT DD DD P W=9.498177e-005 L=4.337074e-007
M2 OUT OUT SS SS N W=5.264844e-006 L=8.382399e-005
```

```
VDD DD 0 DC 6.000000e-001V
VSS SS 0 DC -6.000000e-001V
```

```
.AC DEC 100 100 100MEG
.TRAN 100ns 2000us 1000us
.Trace AC {IOUTDB = DB((I(VOUTB) - I(VOUTA))/2)}
.Trace TRAN {IOUT = I(VOUTB) - I(VOUTA)}
.Trace TRAN I(IINA1) I(IINA2)
.CAPAMIN 1f
```

```
* Dados da Tecnologia
* BSIM3v3.1 parameters for IBM 0.13
* Technology: SCN013
*RUN: T93V (#2_8RF_8LM_DM)
*SPICE 3f5 Level 8, Star-HSPICE Level 49, UTMOST Level 8
.MODEL N NMOS (          LEVEL = 8
+VERSION = 3.1          TNOM = 27          TOX = 3.2E-9
+XJ = 1E-7             NCH = 2.3549E17     VTH0 = 0.0423138
+K1 = 0.3322485       K2 = -0.0298343     K3 = 1E-3
+K3B = 3.6987372      W0 = 1E-7          NLX = 1E-6
+DVT0W = 0            DVT1W = 0          DVT2W = 0
+DVT0 = 1.2848863     DVT1 = 0.1532141     DVT2 = 0.2740377
+U0 = 446.4279855     UA = -1.40947E-10    UB = 3.152185E-18
+UC = 4.34007E-10     VSAT = 7.64383E4     A0 = 0.1112707
+AGS = 0.0734135      B0 = 3.913953E-6     B1 = 5E-6
+KETA = 0.05          A1 = 1.042383E-3     A2 = 0.7518921
+RDSW = 150           PRWG = 0.3496099     PRWB = 0.1112769
+WR = 1               WINT = 2.378741E-10    LINT = 1.040187E-8
+DWG = 1.464878E-8    DWB = 9.306917E-9     VOFF = -0.0240657
+NFACTOR = 2.5        CIT = 0              CDSC = 2.4E-4
+CDSCD = 0            CDSCB = 0             ETA0 = 2.750169E-6
+ETAB = -0.0151402    DSUB = 4.056696E-6    PCLM = 1.9784063
+PDIBLC1 = 0.9657497  PDIBLC2 = 0.01        PDIBLCB = 0.1
+DROUT = 0.9990298    PSCBE1 = 7.957109E10  PSCBE2 = 5.015981E-10
+PVAG = 0.535376      DELTA = 0.01          RSH = 6.4
+MOBMOD = 1           PRT = 0              UTE = -1.5
+KT1 = -0.11          KT1L = 0              KT2 = 0.022
+UA1 = 4.31E-9        UB1 = -7.61E-18       UC1 = -5.6E-11
+AT = 3.3E4           WL = 0                WLN = 1
+WW = 0               WWN = 1               WWL = 0
+LL = 0               LLN = 1               LW = 0
+LWN = 1              LWL = 0               CAPMOD = 2
+XPART = 0.5          CGDO = 3.74E-10       CGSO = 3.74E-10
+CGBO = 1E-12         CJ = 9.581155E-4      PB = 0.9756037
+MJ = 0.4044076      CJSW = 1E-10          PBSW = 0.8002027
+MJSW = 0.6           CJSWG = 3.3E-10       PBSWG = 0.8002027
+MJSWG = 0.6         CF = 0                PVTH0 = -7.55492E-4
+PRDSW = 0            PK2 = 1.510875E-3     WKETA = 0.0128032
+LKETA = 0.0141455    PU0 = -3.9867632     PUA = -3.10294E-11
+PUB = 0              PVSAT = 653.2294237   PETA0 = 1E-4
+PKETA = -9.928283E-3 SAREF = 5.5E-7        SBREF = 5.5E-7
+WLOD = 2E-6          KU0 = -4E-6           KVSAT = 0.2
```

```

+KVTH0 = 2E-8      LLODKU0 = 1.0867072  STIMOD = 2
+WLODKU0 = 1.0990864  LLODVTH = 1      WLODVTH = 1
+LKU0 = 1E-6       WKU0 = 1E-6       LODETA0 = 1
+LKVTH0 = 1.1E-6   WKVTH0 = 1.1E-6   PKVTH0 = 0
+STK2 = 0          LODK2 = 1         STETA0 = 0

```

```

.MODEL P PMOS (                LEVEL = 8)
+VERSION = 3.1      TNOM = 27      TOX = 3.2E-9
+XJ = 1E-7         NCH = 4.1589E17  VTH0 = -0.2176929
+K1 = 0.2566012   K2 = 6.227669E-3  K3 = 9.8203065
+K3B = 20         W0 = 1E-6       NLX = 2.875116E-7
+DVT0W = 0        DVT1W = 0        DVT2W = 0
+DVT0 = 0.1725649  DVT1 = 0.2214064  DVT2 = 0.1
+U0 = 105.8627093  UA = 1.102916E-9  UB = 3.61005E-21
+UC = -4.6651E-11  VSAT = 7.368516E4  A0 = 2
+AGS = 0.7106176  B0 = 4.50136E-6   B1 = 5E-6
+KETA = 0.029826  A1 = 0.1616816   A2 = 0.3
+RDSW = 106.225597  PRWG = -0.4457476  PRWB = 0.5
+WR = 1           WINT = 0         LINT = 1.483182E-8
+DWG = 9.329926E-10  DWB = -3.202354E-8  VOFF = -0.1022829
+NFACTOR = 1.5332272  CIT = 0           CDSC = 2.4E-4
+CDSCD = 0         CDSCB = 0          ETA0 = 0.0115131
+ETAB = -6.181367E-3  DSUB = 3.781207E-3  PCLM = 0.8846898
+PDIBLC1 = 8.605446E-4  PDIBLC2 = 2.244662E-13  PDIBLCB = 0.1
+DROUT = 0         PSCBE1 = 3.388133E9  PSCBE2 = 1.039776E-9
+PVAG = 0         DELTA = 0.01       RSH = 6.4
+MOBMOD = 1       PRT = 0         UTE = -1.5
+KT1 = -0.11     KT1L = 0          KT2 = 0.022
+UA1 = 4.31E-9   UB1 = -7.61E-18  UC1 = -5.6E-11
+AT = 3.3E4      WL = 0           WLN = 1
+WW = 0         WWN = 1          WWL = 0
+LL = 0         LLN = 1          LW = 0
+LWN = 1        LWL = 0          CAPMOD = 2
+XPART = 0.5     CGDO = 3.42E-10   CGSO = 3.42E-10
+CGBO = 1E-12    CJ = 1.155922E-3  PB = 0.8
+MJ = 0.4423239  CJSW = 1.107787E-10  PBSW = 0.8
+MJSWG = 0.1165941  CJSWG = 4.22E-10  PBSWG = 0.8
+MJSWG = 0.1165941  CF = 0           PVTH0 = 4.037191E-4
+PRDSW = 58.2542288  PK2 = 2.351412E-3  WKETA = 0.0350269
+LKETA = 0.0229421  PU0 = -0.8        PUA = -4.64248E-11
+PUB = 0         PVSAT = 50        PETA0 = 1E-4
+PKETA = -0.0142482 )

```

```

.OP EPS_V=1u VMIN=0 VMAX=1.5 DELTAV=150m EPS_I=100p MAXITER=500
HEURISTICS=-4 TRYALL=YES TIMEOUT=-1
.OPS 1u 100m 1n
.H 1fs 1fs 100ns 125m 2
.METHOD GEAR

```

Verificação dos resultados finais:

Verifica restrições de igualdades não lineares ceq(1) e ceq(2)

```

ceq(1):
(Vx-Vdd-VtoP)/nP= 0.0813773934527486
=
-phin*(sqrt(1+x(5)*x(1)/(IsqP*x(3)))-3+log(sqrt(1+x(5)*x(1)/(IsqP*x(3)))-1))1=
0.0813773934527951

ceq(2):
(Vx-Vss-VtoN)/nN= 0.4265409717186366
=
phin*(sqrt(1+x(5)*x(2)/(IsqN*x(4)))-3+log(sqrt(1+x(5)*x(2)/(IsqN*x(4)))-1))= 0.4265409717187008

```

-----  
Verifica os limites (bounds) das variáveis:

0.0000002700000000 <=L1

0.0000004337073861 = L1

0.0001000000000000 >=L1

0.0000002700000000 <=L2

0.0000838239880803 = L2

0.0001000000000000 >=L2

0.0000002700000000 <=W1

0.0000949817678312 = W1

0.0001000000000000 >=W1

0.0000002700000000 <=W2

0.0000052648441848 = W2

0.0001000000000000 >=W2

0.0000000000010000 <= Id

0.0000061342552109 = Id

0.0001000000000000 >= Id

=====  
=====

## EXEMPLO NÚMERO 2

Dado de entrada:

Vx=0.35 Volts

Valores iniciais das variáveis:

L1=1.00000e-004

L2=2.70000e-007

W1=2.70000e-007

W2=2.70000e-007

Id=1.00000e-012

=====  
Resultado das iterações:

=====  
Commercial Ziena License  
KNITRO 7.0.0  
Ziena Optimization, Inc.  
=====

```

algorithm:      1
hessopt:       2
honorbnds:     1
maxit:         50
outlev:        1
KNITRO changing bar_murule from AUTO to 4.
KNITRO changing bar_initpt from AUTO to 2.
KNITRO changing bar_penaltyrule from AUTO to 2.
KNITRO changing bar_penaltycons from AUTO to 1.
KNITRO changing linsolver from AUTO to 5.
KNITRO shifted start point to honor bounds (5 variables).

```

#### Problem Characteristics

```

-----
Objective goal: Minimize
Number of variables:      5
  bounded below:          0
  bounded above:          0
  bounded below and above: 5
  fixed:                   0
  free:                    0
Number of constraints:    2
  linear equalities:      0
  nonlinear equalities:   2
  linear inequalities:    0
  nonlinear inequalities: 0
  range:                  0
Number of nonzeros in Jacobian: 10
Number of nonzeros in Hessian: 15

```

EXIT: Locally optimal solution found.

#### Final Statistics

```

-----
Final objective value      = 4.82514530444625e-010
Final feasibility error (abs / rel) = 1.80e-015 / 1.80e-015
Final optimality error (abs / rel) = 1.17e-008 / 1.17e-008
# of iterations            = 16
# of CG iterations         = 1
# of function evaluations  = 22
# of gradient evaluations  = 17
Total program time (secs) = 0.085 ( 0.090 CPU time)
Time spent in evaluations (secs) = 0.018

```

=====

```

* Arquivo de simulação no SMASH (baseado em ideia de Victor Ariel-UFBA)
* Unidades: metro, Volt, Ampere, Farad, Ohm
* Divisor de Tensão - Modelo ACM
* 25/05/2011 20:44:24
.

```

```

M1 OUT OUT DD DD P W=9.498177e-005 L=4.337074e-007
M2 OUT OUT SS SS N W=5.264844e-006 L=8.382399e-005

```

VDD DD 0 DC 6.000000e-001V  
 VSS SS 0 DC -6.000000e-001V

.AC DEC 100 100 100MEG  
 .TRAN 100ns 2000us 1000us  
 .Trace AC {IOUTDB = DB((I(VOUTB) - I(VOUTA))/2)}  
 .Trace TRAN {IOUT = I(VOUTB) - I(VOUTA)}  
 .Trace TRAN I(IINA1) I(IINA2)  
 .CAPAMIN 1f

\* Dados da Tecnologia

\* BSIM3v3.1 parameters for IBM 0.13

\* Technology: SCN013

\*RUN: T93V (#2\_8RF\_8LM\_DM)

\*SPICE 3f5 Level 8, Star-HSPICE Level 49, UTMOST Level 8

.MODEL N NMOS ( LEVEL = 8  
 +VERSION = 3.1 TNOM = 27 TOX = 3.2E-9  
 +XJ = 1E-7 NCH = 2.3549E17 VTH0 = 0.0423138  
 +K1 = 0.3322485 K2 = -0.0298343 K3 = 1E-3  
 +K3B = 3.6987372 W0 = 1E-7 NLX = 1E-6  
 +DVT0W = 0 DVT1W = 0 DVT2W = 0  
 +DVT0 = 1.2848863 DVT1 = 0.1532141 DVT2 = 0.2740377  
 +U0 = 446.4279855 UA = -1.40947E-10 UB = 3.152185E-18  
 +UC = 4.34007E-10 VSAT = 7.64383E4 A0 = 0.1112707  
 +AGS = 0.0734135 B0 = 3.913953E-6 B1 = 5E-6  
 +KETA = 0.05 A1 = 1.042383E-3 A2 = 0.7518921  
 +RDSW = 150 PRWG = 0.3496099 PRWB = 0.1112769  
 +WR = 1 WINT = 2.378741E-10 LINT = 1.040187E-8  
 +DWG = 1.464878E-8 DWB = 9.306917E-9 VOFF = -0.0240657  
 +NFACTOR = 2.5 CIT = 0 CDSC = 2.4E-4  
 +CDSCD = 0 CDSCB = 0 ETA0 = 2.750169E-6  
 +ETAB = -0.0151402 DSUB = 4.056696E-6 PCLM = 1.9784063  
 +PDIBLC1 = 0.9657497 PDIBLC2 = 0.01 PDIBLCB = 0.1  
 +DROUT = 0.9990298 PSCBE1 = 7.957109E10 PSCBE2 = 5.015981E-10  
 +PVAG = 0.535376 DELTA = 0.01 RSH = 6.4  
 +MOBMOD = 1 PRT = 0 UTE = -1.5  
 +KT1 = -0.11 KT1L = 0 KT2 = 0.022  
 +UA1 = 4.31E-9 UB1 = -7.61E-18 UC1 = -5.6E-11  
 +AT = 3.3E4 WL = 0 WLN = 1  
 +WW = 0 WWN = 1 WWL = 0  
 +LL = 0 LLN = 1 LW = 0  
 +LWN = 1 LWL = 0 CAPMOD = 2  
 +XPART = 0.5 CGDO = 3.74E-10 CGSO = 3.74E-10  
 +CGBO = 1E-12 CJ = 9.581155E-4 PB = 0.9756037  
 +MJ = 0.4044076 CJSW = 1E-10 PBSW = 0.8002027  
 +MJSW = 0.6 CJSWG = 3.3E-10 PBSWG = 0.8002027  
 +MJSWG = 0.6 CF = 0 PVTH0 = -7.55492E-4  
 +PRDSW = 0 PK2 = 1.510875E-3 WKETA = 0.0128032  
 +LKETA = 0.0141455 PU0 = -3.9867632 PUA = -3.10294E-11  
 +PUB = 0 PVSAT = 653.2294237 PETA0 = 1E-4  
 +PKETA = -9.928283E-3 SAREF = 5.5E-7 SBREF = 5.5E-7  
 +WLOD = 2E-6 KU0 = -4E-6 KVSAT = 0.2  
 +KVTH0 = 2E-8 LLODKU0 = 1.0867072 STIMOD = 2  
 +WLODKU0 = 1.0990864 LLODVTH = 1 WLODVTH = 1  
 +LKU0 = 1E-6 WKU0 = 1E-6 LODETA0 = 1

```

+LKVTH0 = 1.1E-6      WKVTH0 = 1.1E-6      PKVTH0 = 0
+STK2 = 0             LODK2 = 1             STETA0 = 0

.MODEL P PMOS (                LEVEL = 8)
+VERSION = 3.1             TNOM = 27             TOX = 3.2E-9
+XJ = 1E-7                NCH = 4.1589E17          VTH0 = -0.2176929
+K1 = 0.2566012          K2 = 6.227669E-3          K3 = 9.8203065
+K3B = 20                W0 = 1E-6                NLX = 2.875116E-7
+DVT0W = 0              DVT1W = 0              DVT2W = 0
+DVT0 = 0.1725649       DVT1 = 0.2214064       DVT2 = 0.1
+U0 = 105.8627093       UA = 1.102916E-9       UB = 3.61005E-21
+UC = -4.6651E-11       VSAT = 7.368516E4      A0 = 2
+AGS = 0.7106176       B0 = 4.50136E-6        B1 = 5E-6
+KETA = 0.029826        A1 = 0.1616816        A2 = 0.3
+RDSW = 106.225597     PRWG = -0.4457476     PRWB = 0.5
+WR = 1                 WINT = 0              LINT = 1.483182E-8
+DWG = 9.329926E-10     DWB = -3.202354E-8     VOFF = -0.1022829
+NFACTOR = 1.5332272    CIT = 0               CDSC = 2.4E-4
+CDSCD = 0              CDSCB = 0             ETA0 = 0.0115131
+ETAB = -6.181367E-3    DSUB = 3.781207E-3    PCLM = 0.8846898
+PDIBLC1 = 8.605446E-4  PDIBLC2 = 2.244662E-13 PDIBLCB = 0.1
+DROUT = 0              PSCBE1 = 3.388133E9   PSCBE2 = 1.039776E-9
+PVAG = 0               DELTA = 0.01          RSH = 6.4
+MOBMOD = 1             PRT = 0              UTE = -1.5
+KT1 = -0.11           KT1L = 0              KT2 = 0.022
+UA1 = 4.31E-9          UB1 = -7.61E-18       UC1 = -5.6E-11
+AT = 3.3E4            WL = 0                WLN = 1
+WW = 0                 WWN = 1               WWL = 0
+LL = 0                 LLN = 1               LW = 0
+LWN = 1                LWL = 0               CAPMOD = 2
+XPART = 0.5           CGDO = 3.42E-10       CGSO = 3.42E-10
+CGBO = 1E-12          CJ = 1.155922E-3      PB = 0.8
+MJ = 0.4423239        CJSW = 1.107787E-10   PBSW = 0.8
+MJSW = 0.1165941      CJSWG = 4.22E-10     PBSWG = 0.8
+MJSWG = 0.1165941     CF = 0                PVTH0 = 4.037191E-4
+PRDSW = 58.2542288     PK2 = 2.351412E-3     WKETA = 0.0350269
+LKETA = 0.0229421      PU0 = -0.8            PUA = -4.64248E-11
+PUB = 0                PVSAT = 50           PETA0 = 1E-4
+PKETA = -0.0142482    )

```

```

.OP EPS_V=1u VMIN=0 VMAX=1.5 DELTAV=150m EPS_I=100p MAXITER=500
HEURISTICS=-4 TRYALL=YES TIMEOUT=-1
.EPS 1u 100m 1n
.H 1fs 1fs 100ns 125m 2
.METHOD GEAR

```

Verificação dos resultados finais:

Verifica restrições de igualdades não lineares ceq(1) e ceq(2)

ceq(1):

$(V_x - V_{dd} - V_{toP})/nP =$

0.0813773934527486

=

$-\text{phit} * (\sqrt{1+x(5)^*x(1)/(I_{sqP}^*x(3))} - 3 + \log(\sqrt{1+x(5)^*x(1)/(I_{sqP}^*x(3))} - 1))1 =$

0.0813773934527951

```

ceq(2):
(Vx-Vss-VtoN)/nN= 0.4265409717186366
=
phit*(sqrt(1+x(5)*x(2)/(IsqN*x(4)))-3+log(sqrt(1+x(5)*x(2)/(IsqN*x(4)))-1))= 0.4265409717187008

```

-----

Verifica os limites (bounds) das variáveis:

```

0.0000002700000000 <=L1
0.0000004337073861 = L1
0.0001000000000000 >=L1

```

```

0.0000002700000000 <=L2
0.0000838239880803 = L2
0.0001000000000000 >=L2

```

```

0.0000002700000000 <=W1
0.0000949817678312 = W1
0.0001000000000000 >=W1

```

```

0.0000002700000000 <=W2
0.0000052648441848 = W2
0.0001000000000000 >=W2

```

```

0.0000000000010000 <= Id
0.0000061342552109 = Id
0.0001000000000000 >= Id

```

=====

=====

### EXEMPLO NÚMERO 3

Dado de entrada:  
Vx=0.35 Volts

Valores iniciais das variáveis:  
L1=2.70000e-007  
L2=1.00000e-004  
W1=2.70000e-007  
W2=2.70000e-007  
Id=1.00000e-012

=====

Resultado das iterações:

=====

Commercial Ziena License  
KNITRO 7.0.0  
Ziena Optimization, Inc.

=====

```

algorithm:      1
hessopt:       2
honorbnds:     1
maxit:         50
outlev:        1
KNITRO changing bar_murule from AUTO to 4.

```

KNITRO changing bar\_initpt from AUTO to 2.  
 KNITRO changing bar\_penaltyrule from AUTO to 2.  
 KNITRO changing bar\_penaltycons from AUTO to 1.  
 KNITRO changing linsolver from AUTO to 5.  
 KNITRO shifted start point to honor bounds (5 variables).

#### Problem Characteristics

```

-----
Objective goal: Minimize
Number of variables:      5
  bounded below:         0
  bounded above:         0
  bounded below and above: 5
  fixed:                 0
  free:                  0
Number of constraints:    2
  linear equalities:      0
  nonlinear equalities:   2
  linear inequalities:    0
  nonlinear inequalities: 0
  range:                 0
Number of nonzeros in Jacobian: 10
Number of nonzeros in Hessian: 15
  
```

EXIT: Locally optimal solution found.

#### Final Statistics

```

-----
Final objective value      = 4.82514530444625e-010
Final feasibility error (abs / rel) = 1.80e-015 / 1.80e-015
Final optimality error (abs / rel) = 1.17e-008 / 1.17e-008
# of iterations            = 16
# of CG iterations         = 1
# of function evaluations  = 22
# of gradient evaluations  = 17
Total program time (secs) = 0.174 ( 0.170 CPU time)
Time spent in evaluations (secs) = 0.036
  
```

```

=====
* Arquivo de simulação no SMASH (baseado em ideia de Victor Ariel-UFBA)
* Unidades: metro, Volt, Ampere, Farad, Ohm
* Divisor de Tensão - Modelo ACM
* 25/05/2011 20:44:24
  
```

```

M1 OUT OUT DD DD P W=9.498177e-005 L=4.337074e-007
M2 OUT OUT SS SS N W=5.264844e-006 L=8.382399e-005
  
```

```

VDD DD 0 DC 6.000000e-001V
VSS SS 0 DC -6.000000e-001V
  
```

```

.AC DEC 100 100 100MEG
.TRAN 100ns 2000us 1000us
.Trace AC {IOUTDB = DB((I(VOUTB) - I(VOUTA))/2)}
.Trace TRAN {IOUT = I(VOUTB) - I(VOUTA)}
  
```



```

+K3B = 20      W0 = 1E-6      NLX = 2.875116E-7
+DVT0W = 0      DVT1W = 0      DVT2W = 0
+DVT0 = 0.1725649  DVT1 = 0.2214064  DVT2 = 0.1
+U0 = 105.8627093  UA = 1.102916E-9  UB = 3.61005E-21
+UC = -4.6651E-11  VSAT = 7.368516E4  A0 = 2
+AGS = 0.7106176  B0 = 4.50136E-6  B1 = 5E-6
+KETA = 0.029826  A1 = 0.1616816  A2 = 0.3
+RDSW = 106.225597  PRWG = -0.4457476  PRWB = 0.5
+WR = 1          WINT = 0          LINT = 1.483182E-8
+DWG = 9.329926E-10  DWB = -3.202354E-8  VOFF = -0.1022829
+NFACTOR = 1.5332272  CIT = 0          CDSC = 2.4E-4
+CDSCD = 0          CDSCB = 0          ETA0 = 0.0115131
+ETAB = -6.181367E-3  DSUB = 3.781207E-3  PCLM = 0.8846898
+PDIBLC1 = 8.605446E-4  PDIBLC2 = 2.244662E-13  PDIBLCB = 0.1
+DROUT = 0          PSCBE1 = 3.388133E9  PSCBE2 = 1.039776E-9
+PVAG = 0          DELTA = 0.01      RSH = 6.4
+MOBMOD = 1          PRT = 0          UTE = -1.5
+KT1 = -0.11        KT1L = 0          KT2 = 0.022
+UA1 = 4.31E-9      UB1 = -7.61E-18  UC1 = -5.6E-11
+AT = 3.3E4         WL = 0           WLN = 1
+WW = 0            WWN = 1          WWL = 0
+LL = 0            LLN = 1          LW = 0
+LWN = 1           LWL = 0          CAPMOD = 2
+XPART = 0.5        CGDO = 3.42E-10  CGSO = 3.42E-10
+CGBO = 1E-12       CJ = 1.155922E-3  PB = 0.8
+MJ = 0.4423239     CJSW = 1.107787E-10  PBSW = 0.8
+MJSW = 0.1165941  CJSWG = 4.22E-10  PBSWG = 0.8
+MJSWG = 0.1165941  CF = 0           PVTH0 = 4.037191E-4
+PRDSW = 58.2542288  PK2 = 2.351412E-3  WKETA = 0.0350269
+LKETA = 0.0229421  PU0 = -0.8       PUA = -4.64248E-11
+PUB = 0           PVSAT = 50       PETA0 = 1E-4
+PKETA = -0.0142482  )

```

```

.OP EPS_V=1u VMIN=0 VMAX=1.5 DELTAV=150m EPS_I=100p MAXITER=500
HEURISTICS=-4 TRYALL=YES TIMEOUT=-1
.OPS 1u 100m 1n
.H 1fs 1fs 100ns 125m 2
.METHOD GEAR

```

Verificação dos resultados finais:

Verifica restrições de igualdades não lineares ceq(1) e ceq(2)

ceq(1):

$(V_x - V_{dd} - V_{toP})/nP =$  0.0813773934527486

=

$-\text{phit} * (\sqrt{(1+x(5)*x(1))/(\text{IsqP}*x(3))}) - 3 + \log(\sqrt{(1+x(5)*x(1))/(\text{IsqP}*x(3))}) - 1) =$

0.0813773934527951

ceq(2):

$(V_x - V_{ss} - V_{toN})/nN =$  0.4265409717186366

=

$\text{phit} * (\sqrt{(1+x(5)*x(2))/(\text{IsqN}*x(4))}) - 3 + \log(\sqrt{(1+x(5)*x(2))/(\text{IsqN}*x(4))}) - 1) =$  0.4265409717187008

-----



KNITRO changing linsolver from AUTO to 5.  
 KNITRO shifted start point to honor bounds (5 variables).

#### Problem Characteristics

```

-----
Objective goal: Minimize
Number of variables:      5
  bounded below:         0
  bounded above:         0
  bounded below and above: 5
  fixed:                  0
  free:                   0
Number of constraints:    2
  linear equalities:      0
  nonlinear equalities:   2
  linear inequalities:     0
  nonlinear inequalities: 0
  range:                  0
Number of nonzeros in Jacobian: 10
Number of nonzeros in Hessian: 15
  
```

EXIT: Locally optimal solution found.

#### Final Statistics

```

-----
Final objective value      = 4.82514530444625e-010
Final feasibility error (abs / rel) = 1.80e-015 / 1.80e-015
Final optimality error (abs / rel) = 1.17e-008 / 1.17e-008
# of iterations            = 16
# of CG iterations         = 1
# of function evaluations  = 22
# of gradient evaluations  = 17
Total program time (secs) = 0.226 ( 0.220 CPU time)
Time spent in evaluations (secs) = 0.083
  
```

```

=====
* Arquivo de simulação no SMASH (baseado em ideia de Victor Ariel-UFBA)
* Unidades: metro, Volt, Ampere, Farad, Ohm
* Divisor de Tensão - Modelo ACM
* 25/05/2011 20:37:28
  
```

```

M1 OUT OUT DD DD P W=9.498177e-005 L=4.337074e-007
M2 OUT OUT SS SS N W=5.264844e-006 L=8.382399e-005
  
```

```

VDD DD 0 DC 6.000000e-001V
VSS SS 0 DC -6.000000e-001V
  
```

```

.AC DEC 100 100 100MEG
.TRAN 100ns 2000us 1000us
.Trace AC {IOUTDB = DB((I(VOUTB) - I(VOUTA))/2)}
.Trace TRAN {IOUT = I(VOUTB) - I(VOUTA)}
.Trace TRAN I(IINA1) I(IINA2)
.CAPAMIN 1f
  
```

```

* Dados da Tecnologia
* BSIM3v3.1 parameters for IBM 0.13
* Technology: SCN013
*RUN: T93V (#2_8RF_8LM_DM)
*SPICE 3f5 Level 8, Star-HSPICE Level 49, UTMOST Level 8
.MODEL N NMOS (          LEVEL = 8
+VERSION = 3.1          TNOM = 27          TOX = 3.2E-9
+XJ = 1E-7             NCH = 2.3549E17     VTH0 = 0.0423138
+K1 = 0.3322485       K2 = -0.0298343     K3 = 1E-3
+K3B = 3.6987372      W0 = 1E-7          NLX = 1E-6
+DVT0W = 0            DVT1W = 0            DVT2W = 0
+DVT0 = 1.2848863     DVT1 = 0.1532141     DVT2 = 0.2740377
+U0 = 446.4279855     UA = -1.40947E-10    UB = 3.152185E-18
+UC = 4.34007E-10     VSAT = 7.64383E4     A0 = 0.1112707
+AGS = 0.0734135     B0 = 3.913953E-6     B1 = 5E-6
+KETA = 0.05          A1 = 1.042383E-3     A2 = 0.7518921
+RDSW = 150           PRWG = 0.3496099     PRWB = 0.1112769
+WR = 1               WINT = 2.378741E-10 LINT = 1.040187E-8
+DWG = 1.464878E-8    DWB = 9.306917E-9    VOFF = -0.0240657
+NFACTOR = 2.5        CIT = 0              CDSC = 2.4E-4
+CDSCD = 0            CDSCB = 0             ETA0 = 2.750169E-6
+ETAB = -0.0151402    DSUB = 4.056696E-6   PCLM = 1.9784063
+PDIBLC1 = 0.9657497 PDIBLC2 = 0.01        PDIBLCB = 0.1
+DROUT = 0.9990298    PSCBE1 = 7.957109E10 PSCBE2 = 5.015981E-10
+PVAG = 0.535376     DELTA = 0.01         RSH = 6.4
+MOBMOD = 1          PRT = 0              UTE = -1.5
+KT1 = -0.11         KT1L = 0             KT2 = 0.022
+UA1 = 4.31E-9        UB1 = -7.61E-18      UC1 = -5.6E-11
+AT = 3.3E4           WL = 0               WLN = 1
+WW = 0              WWN = 1              WWL = 0
+LL = 0              LLN = 1              LW = 0
+LWN = 1             LWL = 0              CAPMOD = 2
+XPART = 0.5         CGDO = 3.74E-10      CGSO = 3.74E-10
+CGBO = 1E-12        CJ = 9.581155E-4     PB = 0.9756037
+MJ = 0.4044076      CJSW = 1E-10         PBSW = 0.8002027
+MJSW = 0.6          CJSWG = 3.3E-10      PBSWG = 0.8002027
+MJSWG = 0.6         CF = 0               PVTH0 = -7.55492E-4
+PRDSW = 0           PK2 = 1.510875E-3    WKETA = 0.0128032
+LKETA = 0.0141455   PU0 = -3.9867632     PUA = -3.10294E-11
+PUB = 0             PVSAT = 653.2294237 PETA0 = 1E-4
+PKETA = -9.928283E-3 SAREF = 5.5E-7       SBREF = 5.5E-7
+WLOD = 2E-6         KU0 = -4E-6          KVSAT = 0.2
+KVTH0 = 2E-8        LLODKU0 = 1.0867072 STIMOD = 2
+WLODKU0 = 1.0990864 LLODVTH = 1          WLODVTH = 1
+LKU0 = 1E-6         WKU0 = 1E-6          LODETA0 = 1
+LKVTH0 = 1.1E-6     WKVTH0 = 1.1E-6     PKVTH0 = 0
+STK2 = 0            LODK2 = 1            STETA0 = 0

.MODEL P PMOS (          LEVEL = 8)
+VERSION = 3.1          TNOM = 27          TOX = 3.2E-9
+XJ = 1E-7             NCH = 4.1589E17     VTH0 = -0.2176929
+K1 = 0.2566012       K2 = 6.227669E-3     K3 = 9.8203065
+K3B = 20             W0 = 1E-6          NLX = 2.875116E-7
+DVT0W = 0            DVT1W = 0            DVT2W = 0
+DVT0 = 0.1725649     DVT1 = 0.2214064     DVT2 = 0.1

```

```

+U0 = 105.8627093 UA = 1.102916E-9 UB = 3.61005E-21
+UC = -4.6651E-11 VSAT = 7.368516E4 A0 = 2
+AGS = 0.7106176 B0 = 4.50136E-6 B1 = 5E-6
+KETA = 0.029826 A1 = 0.1616816 A2 = 0.3
+RDSW = 106.225597 PRWG = -0.4457476 PRWB = 0.5
+WR = 1 WINT = 0 LINT = 1.483182E-8
+DWG = 9.329926E-10 DWB = -3.202354E-8 VOFF = -0.1022829
+NFACTOR = 1.5332272 CIT = 0 CDSC = 2.4E-4
+CDSCD = 0 CDSCB = 0 ETA0 = 0.0115131
+ETAB = -6.181367E-3 DSUB = 3.781207E-3 PCLM = 0.8846898
+PDIBLC1 = 8.605446E-4 PDIBLC2 = 2.244662E-13 PDIBLCB = 0.1
+DROUT = 0 PSCBE1 = 3.388133E9 PSCBE2 = 1.039776E-9
+PVAG = 0 DELTA = 0.01 RSH = 6.4
+MOBMOD = 1 PRT = 0 UTE = -1.5
+KT1 = -0.11 KT1L = 0 KT2 = 0.022
+UA1 = 4.31E-9 UB1 = -7.61E-18 UC1 = -5.6E-11
+AT = 3.3E4 WL = 0 WLN = 1
+WW = 0 WWN = 1 WWL = 0
+LL = 0 LLN = 1 LW = 0
+LWN = 1 LWL = 0 CAPMOD = 2
+XPART = 0.5 CGDO = 3.42E-10 CGSO = 3.42E-10
+CGBO = 1E-12 CJ = 1.155922E-3 PB = 0.8
+MJ = 0.4423239 CJSW = 1.107787E-10 PBSW = 0.8
+MJSW = 0.1165941 CJSWG = 4.22E-10 PBSWG = 0.8
+MJSWG = 0.1165941 CF = 0 PVTH0 = 4.037191E-4
+PRDSW = 58.2542288 PK2 = 2.351412E-3 WKETA = 0.0350269
+LKETA = 0.0229421 PU0 = -0.8 PUA = -4.64248E-11
+PUB = 0 PVSAT = 50 PETA0 = 1E-4
+PKETA = -0.0142482 )

```

```

.OP EPS_V=1u VMIN=0 VMAX=1.5 DELTAV=150m EPS_I=100p MAXITER=500
HEURISTICS=-4 TRYALL=YES TIMEOUT=-1
.EPS 1u 100m 1n
.H 1fs 1fs 100ns 125m 2
.METHOD GEAR

```

Verificação dos resultados finais:

Verifica restrições de igualdades não lineares ceq(1) e ceq(2)

ceq(1):

$(V_x - V_{dd} - V_{toP})/n_P = 0.0813773934527486$

=

$-\text{phit}(\sqrt{(1+x(5)^x(1))/(I_{sqP}^x(3))}) - 3 + \log(\sqrt{(1+x(5)^x(1))/(I_{sqP}^x(3))}) - 1) =$

0.0813773934527951

ceq(2):

$(V_x - V_{ss} - V_{toN})/n_N = 0.4265409717186366$

=

$\text{phit}(\sqrt{(1+x(5)^x(2))/(I_{sqN}^x(4))}) - 3 + \log(\sqrt{(1+x(5)^x(2))/(I_{sqN}^x(4))}) - 1) = 0.4265409717187008$

-----

Verifica os limites (bounds) das variáveis:

0.0000002700000000 <=L1  
 0.0000004337073861 = L1  
 0.0001000000000000 >=L1

0.0000002700000000 <=L2  
 0.0000838239880803 = L2  
 0.0001000000000000 >=L2

0.0000002700000000 <=W1  
 0.0000949817678312 = W1  
 0.0001000000000000 >=W1

0.0000002700000000 <=W2  
 0.0000052648441848 = W2  
 0.0001000000000000 >=W2

0.0000000000010000 <= Id  
 0.0000061342552109 = Id  
 0.0001000000000000 >= Id

=====  
 =====

#### EXEMPLO NÚMERO 31

Dado de entrada:

Vx=0.35 Volts

Valores iniciais das variáveis:

L1=2.70000e-007  
 L2=1.00000e-004  
 W1=1.00000e-004  
 W2=1.00000e-004  
 Id=1.00000e-004

=====  
 =====

Resultado das iterações:

=====

Commercial Ziena License  
 KNITRO 7.0.0  
 Ziena Optimization, Inc.

=====

algorithm: 1  
 hessopt: 2  
 honorbnds: 1  
 maxit: 50  
 outlev: 1  
 KNITRO changing bar\_murule from AUTO to 4.  
 KNITRO changing bar\_initpt from AUTO to 2.  
 KNITRO changing bar\_penaltyrule from AUTO to 2.  
 KNITRO changing bar\_penaltycons from AUTO to 1.  
 KNITRO changing linsolver from AUTO to 5.  
 KNITRO shifted start point to honor bounds (5 variables).

### Problem Characteristics

```

-----
Objective goal: Minimize
Number of variables:      5
  bounded below:         0
  bounded above:         0
  bounded below and above: 5
  fixed:                  0
  free:                   0
Number of constraints:    2
  linear equalities:      0
  nonlinear equalities:   2
  linear inequalities:    0
  nonlinear inequalities: 0
  range:                  0
Number of nonzeros in Jacobian: 10
Number of nonzeros in Hessian: 15

```

EXIT: Locally optimal solution found.

### Final Statistics

```

-----
Final objective value      = 4.82514530444625e-010
Final feasibility error (abs / rel) = 1.80e-015 / 1.80e-015
Final optimality error (abs / rel) = 1.17e-008 / 1.17e-008
# of iterations            = 16
# of CG iterations         = 1
# of function evaluations  = 22
# of gradient evaluations  = 17
Total program time (secs) = 0.194 ( 0.200 CPU time)
Time spent in evaluations (secs) = 0.082

```

```

=====
* Arquivo de simulação no SMASH (baseado em ideia de Victor Ariel-UFBA)
* Unidades: metro, Volt, Ampere, Farad, Ohm
* Divisor de Tensão - Modelo ACM
* 25/05/2011 20:37:28

```

```

M1 OUT OUT DD DD P W=9.498177e-005 L=4.337074e-007
M2 OUT OUT SS SS N W=5.264844e-006 L=8.382399e-005

```

```

VDD DD 0 DC 6.000000e-001V
VSS SS 0 DC -6.000000e-001V

```

```

.AC DEC 100 100 100MEG
.TRAN 100ns 2000us 1000us
.Trace AC {IOUTDB = DB((I(VOUTB) - I(VOUTA))/2)}
.Trace TRAN {IOUT = I(VOUTB) - I(VOUTA)}
.Trace TRAN I(IINA1) I(IINA2)
.CAPAMIN 1f

```

```

* Dados da Tecnologia
* BSIM3v3.1 parameters for IBM 0.13
* Technology: SCN013

```

```

*RUN: T93V (#2_8RF_8LM_DM)
*SPICE 3f5 Level 8, Star-HSPICE Level 49, UTMOST Level 8
.MODEL N NMOS (                LEVEL = 8
+VERSION = 3.1      TNOM = 27      TOX = 3.2E-9
+XJ = 1E-7         NCH = 2.3549E17  VTH0 = 0.0423138
+K1 = 0.3322485   K2 = -0.0298343  K3 = 1E-3
+K3B = 3.6987372  W0 = 1E-7       NLX = 1E-6
+DVT0W = 0        DVT1W = 0       DVT2W = 0
+DVT0 = 1.2848863  DVT1 = 0.1532141  DVT2 = 0.2740377
+U0 = 446.4279855  UA = -1.40947E-10  UB = 3.152185E-18
+UC = 4.34007E-10  VSAT = 7.64383E4   A0 = 0.1112707
+AGS = 0.0734135  B0 = 3.913953E-6  B1 = 5E-6
+KETA = 0.05      A1 = 1.042383E-3  A2 = 0.7518921
+RDSW = 150      PRWG = 0.3496099  PRWB = 0.1112769
+WR = 1          WINT = 2.378741E-10  LINT = 1.040187E-8
+DWG = 1.464878E-8  DWB = 9.306917E-9  VOFF = -0.0240657
+NFACTOR = 2.5     CIT = 0         CDSC = 2.4E-4
+CDSCD = 0        CDSCB = 0       ETA0 = 2.750169E-6
+ETAB = -0.0151402  DSUB = 4.056696E-6  PCLM = 1.9784063
+PDIBLC1 = 0.9657497  PDIBLC2 = 0.01     PDIBLCB = 0.1
+DROUT = 0.9990298  PSCBE1 = 7.957109E10  PSCBE2 = 5.015981E-10
+PVAG = 0.535376   DELTA = 0.01       RSH = 6.4
+MOBMOD = 1        PRT = 0         UTE = -1.5
+KT1 = -0.11      KT1L = 0         KT2 = 0.022
+UA1 = 4.31E-9    UB1 = -7.61E-18  UC1 = -5.6E-11
+AT = 3.3E4       WL = 0         WLN = 1
+WW = 0           WWN = 1        WWL = 0
+LL = 0           LLN = 1        LW = 0
+LWN = 1          LWL = 0        CAPMOD = 2
+XPART = 0.5      CGDO = 3.74E-10   CGSO = 3.74E-10
+CGBO = 1E-12     CJ = 9.581155E-4  PB = 0.9756037
+MJ = 0.4044076   CJSW = 1E-10     PBSW = 0.8002027
+MJSW = 0.6       CJSWG = 3.3E-10  PBSWG = 0.8002027
+MJSWG = 0.6      CF = 0           PVTH0 = -7.55492E-4
+PRDSW = 0        PK2 = 1.510875E-3  WKETA = 0.0128032
+LKETA = 0.0141455  PU0 = -3.9867632  PUA = -3.10294E-11
+PUB = 0          PVSAT = 653.2294237  PETA0 = 1E-4
+PKETA = -9.928283E-3  SAREF = 5.5E-7  SBREF = 5.5E-7
+WLOD = 2E-6      KU0 = -4E-6     KVSAT = 0.2
+KVTH0 = 2E-8     LLODKU0 = 1.0867072  STIMOD = 2
+WLODKU0 = 1.0990864  LLODVTH = 1     WLODVTH = 1
+LKU0 = 1E-6      WKU0 = 1E-6     LODETA0 = 1
+LKVTH0 = 1.1E-6  WKVTH0 = 1.1E-6  PKVTH0 = 0
+STK2 = 0         LODK2 = 1        STETA0 = 0

.MODEL P PMOS (                LEVEL = 8)
+VERSION = 3.1      TNOM = 27      TOX = 3.2E-9
+XJ = 1E-7         NCH = 4.1589E17  VTH0 = -0.2176929
+K1 = 0.2566012   K2 = 6.227669E-3  K3 = 9.8203065
+K3B = 20         W0 = 1E-6       NLX = 2.875116E-7
+DVT0W = 0        DVT1W = 0       DVT2W = 0
+DVT0 = 0.1725649  DVT1 = 0.2214064  DVT2 = 0.1
+U0 = 105.8627093  UA = 1.102916E-9  UB = 3.61005E-21
+UC = -4.6651E-11  VSAT = 7.368516E4  A0 = 2
+AGS = 0.7106176  B0 = 4.50136E-6  B1 = 5E-6

```

```

+KETA = 0.029826  A1 = 0.1616816  A2 = 0.3
+RDSW = 106.225597  PRWG = -0.4457476  PRWB = 0.5
+WR = 1  WINT = 0  LINT = 1.483182E-8
+DWG = 9.329926E-10  DWB = -3.202354E-8  VOFF = -0.1022829
+NFACTOR = 1.5332272  CIT = 0  CDSC = 2.4E-4
+CDSCD = 0  CDSCB = 0  ETA0 = 0.0115131
+ETAB = -6.181367E-3  DSUB = 3.781207E-3  PCLM = 0.8846898
+PDIBLC1 = 8.605446E-4  PDIBLC2 = 2.244662E-13  PDIBLCB = 0.1
+DROUT = 0  PSCBE1 = 3.388133E9  PSCBE2 = 1.039776E-9
+PVAG = 0  DELTA = 0.01  RSH = 6.4
+MOBMOD = 1  PRT = 0  UTE = -1.5
+KT1 = -0.11  KT1L = 0  KT2 = 0.022
+UA1 = 4.31E-9  UB1 = -7.61E-18  UC1 = -5.6E-11
+AT = 3.3E4  WL = 0  WLN = 1
+WW = 0  WWN = 1  WWL = 0
+LL = 0  LLN = 1  LW = 0
+LWN = 1  LWL = 0  CAPMOD = 2
+XPART = 0.5  CGDO = 3.42E-10  CGSO = 3.42E-10
+CGBO = 1E-12  CJ = 1.155922E-3  PB = 0.8
+MJ = 0.4423239  CJSW = 1.107787E-10  PBSW = 0.8
+MJSW = 0.1165941  CJSWG = 4.22E-10  PBSWG = 0.8
+MJSWG = 0.1165941  CF = 0  PVTH0 = 4.037191E-4
+PRDSW = 58.2542288  PK2 = 2.351412E-3  WKETA = 0.0350269
+LKETA = 0.0229421  PU0 = -0.8  PUA = -4.64248E-11
+PUB = 0  PVSAT = 50  PETA0 = 1E-4
+PKETA = -0.0142482  )

```

```

.OP EPS_V=1u VMIN=0 VMAX=1.5 DELTAV=150m EPS_I=100p MAXITER=500
HEURISTICS=-4 TRYALL=YES TIMEOUT=-1
.EPS 1u 100m 1n
.H 1fs 1fs 100ns 125m 2
.METHOD GEAR

```

Verificação dos resultados finais:

Verifica restrições de igualdades não lineares ceq(1) e ceq(2)

ceq(1):

$(V_x - V_{dd} - V_{toP})/nP = 0.0813773934527486$

=

$-\text{phit} * (\sqrt{(1+x(5)^*x(1))/(\text{IsqP}^*x(3))}) - 3 + \log(\sqrt{(1+x(5)^*x(1))/(\text{IsqP}^*x(3))} - 1)) = 0.0813773934527951$

ceq(2):

$(V_x - V_{ss} - V_{toN})/nN = 0.4265409717186366$

=

$\text{phit} * (\sqrt{(1+x(5)^*x(2))/(\text{IsqN}^*x(4))}) - 3 + \log(\sqrt{(1+x(5)^*x(2))/(\text{IsqN}^*x(4))} - 1)) = 0.4265409717187008$

-----

Verifica os limites (bounds) das variáveis:

0.0000002700000000 <=L1  
 0.0000004337073861 = L1  
 0.0001000000000000 >=L1

0.0000002700000000 <=L2  
 0.0000838239880803 = L2  
 0.0001000000000000 >=L2

0.0000002700000000 <=W1  
 0.0000949817678312 = W1  
 0.0001000000000000 >=W1

0.0000002700000000 <=W2  
 0.0000052648441848 = W2  
 0.0001000000000000 >=W2

0.0000000000010000 <= Id  
 0.0000061342552109 = Id  
 0.0001000000000000 >= Id

=====  
 =====

### EXEMPLO NÚMERO 32

Dado de entrada:

Vx=0.35 Volts

Valores iniciais das variáveis:

L1=1.00000e-004  
 L2=1.00000e-004  
 W1=1.00000e-004  
 W2=1.00000e-004  
 Id=1.00000e-004

=====  
 =====

Resultado das iterações:

=====

Commercial Ziena License  
 KNITRO 7.0.0  
 Ziena Optimization, Inc.

=====

algorithm: 1  
 hessopt: 2  
 honorbnds: 1  
 maxit: 50  
 outlev: 1  
 KNITRO changing bar\_murule from AUTO to 4.  
 KNITRO changing bar\_initpt from AUTO to 2.  
 KNITRO changing bar\_penaltyrule from AUTO to 2.  
 KNITRO changing bar\_penaltycons from AUTO to 1.  
 KNITRO changing linsolver from AUTO to 5.  
 KNITRO shifted start point to honor bounds (5 variables).

### Problem Characteristics

```

-----
Objective goal: Minimize
Number of variables:      5
  bounded below:         0
  bounded above:         0
  bounded below and above: 5
  fixed:                 0
  free:                  0
Number of constraints:    2
  linear equalities:      0
  nonlinear equalities:   2
  linear inequalities:     0
  nonlinear inequalities:  0
  range:                 0
Number of nonzeros in Jacobian: 10
Number of nonzeros in Hessian: 15

```

EXIT: Locally optimal solution found.

### Final Statistics

```

-----
Final objective value      = 4.82514530444625e-010
Final feasibility error (abs / rel) = 1.80e-015 / 1.80e-015
Final optimality error (abs / rel) = 1.17e-008 / 1.17e-008
# of iterations           = 16
# of CG iterations        = 1
# of function evaluations = 22
# of gradient evaluations = 17
Total program time (secs) = 0.139 ( 0.140 CPU time)
Time spent in evaluations (secs) = 0.031

```

```

=====
* Arquivo de simulação no SMASH (baseado em ideia de Victor Ariel-UFBA)
* Unidades: metro, Volt, Ampere, Farad, Ohm
* Divisor de Tensão - Modelo ACM
* 25/05/2011 20:37:29

```

```

M1 OUT OUT DD DD P W=9.498177e-005 L=4.337074e-007
M2 OUT OUT SS SS N W=5.264844e-006 L=8.382399e-005

```

```

VDD DD 0 DC 6.000000e-001V
VSS SS 0 DC -6.000000e-001V

```

```

.AC DEC 100 100 100MEG
.TRAN 100ns 2000us 1000us
.Trace AC {IOUTDB = DB((I(VOUTB) - I(VOUTA))/2)}
.Trace TRAN {IOUT = I(VOUTB) - I(VOUTA)}
.Trace TRAN I(IINA1) I(IINA2)
.CAPAMIN 1f

```

```

* Dados da Tecnologia
* BSIM3v3.1 parameters for IBM 0.13
* Technology: SCN013

```

```

*RUN: T93V (#2_8RF_8LM_DM)
*SPICE 3f5 Level 8, Star-HSPICE Level 49, UTMOST Level 8
.MODEL N NMOS (                LEVEL = 8
+VERSION = 3.1      TNOM = 27      TOX = 3.2E-9
+XJ = 1E-7         NCH = 2.3549E17  VTH0 = 0.0423138
+K1 = 0.3322485   K2 = -0.0298343  K3 = 1E-3
+K3B = 3.6987372  W0 = 1E-7       NLX = 1E-6
+DVT0W = 0        DVT1W = 0       DVT2W = 0
+DVT0 = 1.2848863  DVT1 = 0.1532141  DVT2 = 0.2740377
+U0 = 446.4279855  UA = -1.40947E-10  UB = 3.152185E-18
+UC = 4.34007E-10  VSAT = 7.64383E4   A0 = 0.1112707
+AGS = 0.0734135  B0 = 3.913953E-6  B1 = 5E-6
+KETA = 0.05      A1 = 1.042383E-3  A2 = 0.7518921
+RDSW = 150      PRWG = 0.3496099  PRWB = 0.1112769
+WR = 1          WINT = 2.378741E-10  LINT = 1.040187E-8
+DWG = 1.464878E-8  DWB = 9.306917E-9  VOFF = -0.0240657
+NFACTOR = 2.5     CIT = 0         CDSC = 2.4E-4
+CDSCD = 0        CDSCB = 0       ETA0 = 2.750169E-6
+ETAB = -0.0151402  DSUB = 4.056696E-6  PCLM = 1.9784063
+PDIBLC1 = 0.9657497  PDIBLC2 = 0.01     PDIBLCB = 0.1
+DROUT = 0.9990298  PSCBE1 = 7.957109E10  PSCBE2 = 5.015981E-10
+PVAG = 0.535376   DELTA = 0.01       RSH = 6.4
+MOBMOD = 1       PRT = 0         UTE = -1.5
+KT1 = -0.11     KT1L = 0         KT2 = 0.022
+UA1 = 4.31E-9   UB1 = -7.61E-18  UC1 = -5.6E-11
+AT = 3.3E4     WL = 0         WLN = 1
+WW = 0        WWN = 1       WWL = 0
+LL = 0        LLN = 1       LW = 0
+LWN = 1       LWL = 0       CAPMOD = 2
+XPART = 0.5    CGDO = 3.74E-10  CGSO = 3.74E-10
+CGBO = 1E-12   CJ = 9.581155E-4  PB = 0.9756037
+MJ = 0.4044076  CJSW = 1E-10     PBSW = 0.8002027
+MJSW = 0.6     CJSWG = 3.3E-10  PBSWG = 0.8002027
+MJSWG = 0.6    CF = 0         PVTH0 = -7.55492E-4
+PRDSW = 0      PK2 = 1.510875E-3  WKETA = 0.0128032
+LKETA = 0.0141455  PU0 = -3.9867632  PUA = -3.10294E-11
+PUB = 0        PVSAT = 653.2294237  PETA0 = 1E-4
+PKETA = -9.928283E-3  SAREF = 5.5E-7  SBREF = 5.5E-7
+WLOD = 2E-6    KU0 = -4E-6    KVSAT = 0.2
+KVTH0 = 2E-8   LLODKU0 = 1.0867072  STIMOD = 2
+WLODKU0 = 1.0990864  LLODVTH = 1     WLODVTH = 1
+LKU0 = 1E-6    WKU0 = 1E-6     LODETA0 = 1
+LKVTH0 = 1.1E-6  WKVTH0 = 1.1E-6  PKVTH0 = 0
+STK2 = 0       LODK2 = 1     STETA0 = 0

.MODEL P PMOS (                LEVEL = 8)
+VERSION = 3.1      TNOM = 27      TOX = 3.2E-9
+XJ = 1E-7         NCH = 4.1589E17  VTH0 = -0.2176929
+K1 = 0.2566012   K2 = 6.227669E-3  K3 = 9.8203065
+K3B = 20         W0 = 1E-6       NLX = 2.875116E-7
+DVT0W = 0        DVT1W = 0       DVT2W = 0
+DVT0 = 0.1725649  DVT1 = 0.2214064  DVT2 = 0.1
+U0 = 105.8627093  UA = 1.102916E-9  UB = 3.61005E-21
+UC = -4.6651E-11  VSAT = 7.368516E4  A0 = 2
+AGS = 0.7106176  B0 = 4.50136E-6  B1 = 5E-6

```

```

+KETA = 0.029826   A1   = 0.1616816   A2   = 0.3
+RDSW = 106.225597 PRWG = -0.4457476 PRWB = 0.5
+WR   = 1          WINT = 0          LINT = 1.483182E-8
+DWG  = 9.329926E-10 DWB  = -3.202354E-8 VOFF = -0.1022829
+NFACTOR = 1.5332272 CIT   = 0          CDSC = 2.4E-4
+CDSCD = 0          CDSCB = 0          ETA0 = 0.0115131
+ETAB = -6.181367E-3 DSUB = 3.781207E-3 PCLM = 0.8846898
+PDIBLC1 = 8.605446E-4 PDIBLC2 = 2.244662E-13 PDIBLCB = 0.1
+DROUT = 0          PSCBE1 = 3.388133E9 PSCBE2 = 1.039776E-9
+PVAG = 0          DELTA = 0.01       RSH  = 6.4
+MOBMOD = 1        PRT   = 0          UTE  = -1.5
+KT1   = -0.11     KT1L = 0          KT2  = 0.022
+UA1   = 4.31E-9   UB1   = -7.61E-18 UC1   = -5.6E-11
+AT    = 3.3E4     WL    = 0          WLN  = 1
+WW    = 0         WWN   = 1          WWL  = 0
+LL    = 0         LLN   = 1          LW   = 0
+LWN   = 1         LWL   = 0          CAPMOD = 2
+XPART = 0.5       CGDO  = 3.42E-10 CGSO  = 3.42E-10
+CGBO  = 1E-12     CJ    = 1.155922E-3 PB   = 0.8
+MJ    = 0.4423239 CJSW  = 1.107787E-10 PBSW = 0.8
+MJSW  = 0.1165941 CJSWG = 4.22E-10 PBSWG = 0.8
+MJSWG = 0.1165941 CF    = 0          PVTH0 = 4.037191E-4
+PRDSW = 58.2542288 PK2   = 2.351412E-3 WKETA = 0.0350269
+LKETA = 0.0229421 PU0   = -0.8       PUA   = -4.64248E-11
+PUB   = 0         PVSAT = 50          PETA0 = 1E-4
+PKETA = -0.0142482 )

```

```

.OP EPS_V=1u VMIN=0 VMAX=1.5 DELTAV=150m EPS_I=100p MAXITER=500
HEURISTICS=-4 TRYALL=YES TIMEOUT=-1
.EPS 1u 100m 1n
.H 1fs 1fs 100ns 125m 2
.METHOD GEAR

```

Verificação dos resultados finais:

Verifica restrições de igualdades não lineares ceq(1) e ceq(2)

ceq(1):

$(V_x - V_{dd} - V_{toP})/n_P =$  0.0813773934527486

=

$-\text{phit}(\sqrt{(1+x(5)*x(1))/(I_{sqP}*x(3))})-3+\log(\sqrt{(1+x(5)*x(1))/(I_{sqP}*x(3))}-1))1=$

0.0813773934527951

ceq(2):

$(V_x - V_{ss} - V_{toN})/n_N =$  0.4265409717186366

=

$\text{phit}(\sqrt{(1+x(5)*x(2))/(I_{sqN}*x(4))})-3+\log(\sqrt{(1+x(5)*x(2))/(I_{sqN}*x(4))}-1))=$  0.4265409717187008

-----

Verifica os limites (bounds) das variáveis:

0.0000002700000000 <=L1  
0.0000004337073861 = L1  
0.0001000000000000 >=L1

0.0000002700000000 <=L2  
0.0000838239880803 = L2  
0.0001000000000000 >=L2

0.0000002700000000 <=W1  
0.0000949817678312 = W1  
0.0001000000000000 >=W1

0.0000002700000000 <=W2  
0.0000052648441848 = W2  
0.0001000000000000 >=W2

0.0000000000010000 <= Id  
0.0000061342552109 = Id  
0.0001000000000000 >= Id

=====  
=====

## C.2 Projeto do Integrador de Seevinck (Exemplos A, B e C)

### EXEMPLO A

OTIMIZAÇÃO DE CIRCUITOS INTEGRADOS - MATLAB OPTIMIZATION TOOLBOX -  
KNITRO

MÉTODOS DOS PONTOS INTERIORES

Digite o valor de "Kint" em radianos por segundo:

?  $2\pi \cdot 10^5$

Digite o valor de "iinmax" em Amperes:

?  $2e-7$

Digite o valor de "iinmin" em Amperes:

?  $-2e-7$

Defina a quantidade máxima de iterações do KNITRO:

? 50

Os valores "default" das tolerâncias são:

TolX=  $1e-15$  (variáveis)

TolFun= $1e-06$  (função objetivo)

TolCon= $1e-06$  (região factível das restrições)

Escolha se quer mudar as tolerâncias.

Digite "S" ou "N": N

=====

Cálculo da Otimização com o MATLAB Optimization Toolbox/KNITRO  
Método da Otimização: Pontos Interiores

Dados de entrada:

Kint= $628318.5307$  radianos por segundo

iinmax= $2e-007$  Amperes

iinmin= $-2e-007$  Amperes

Valores iniciais das variáveis:

I= $2.26000e-007$  lo= $2.26000e-007$  lbias= $5.00000e-007$

L= $5.00000e-007$  W1=W2= $1.00000e-005$  W3= $1.00000e-005$

W4= $1.00000e-005$  W6=W11= $5.00000e-006$  W7=W8= $5.00000e-006$

W10= $1.00000e-005$  C= $2.00000e-011$

Limites ("bounds") inferiores e superiores:

(Unidades: Amperes, metros e Farads)

$1.000e-012 \leq I \leq 1.000e-005$

$1.000e-012 \leq lo \leq 1.000e-005$

$1.000e-012 \leq lbias \leq 1.000e-005$

$2.600e-007 \leq L \leq 1.000e-004$

$2.600e-007 \leq W1=W2 \leq 1.000e-004$

$2.600e-007 \leq W3 \leq 1.000e-004$

$2.600e-007 \leq W4 \leq 1.000e-004$

$2.600e-007 \leq W6=W11 \leq 1.000e-004$

$2.600e-007 \leq W7=W8 \leq 1.000e-004$

$2.600e-007 \leq W10 \leq 1.000e-004$

$1.000e-011 \leq C \leq 1.000e-010$

Tolerâncias especificadas:

---

TolX= 1e-015 (variáveis)  
 TolFun=1e-006 (função objetivo)  
 TolCon=1e-006 (região factível das restrições)

=====  
 Resultado das iterações:

=====  
 Commercial Ziena License  
 KNITRO 7.0.0  
 Ziena Optimization, Inc.  
 =====

algorithm: 1  
 hessopt: 2  
 honorbnds: 1  
 maxit: 50  
 outlev: 4  
 KNITRO changing bar\_murule from AUTO to 4.  
 KNITRO changing bar\_initpt from AUTO to 2.  
 KNITRO changing bar\_penaltyrule from AUTO to 2.  
 KNITRO changing bar\_penaltycons from AUTO to 1.  
 KNITRO changing linsolver from AUTO to 5.  
 KNITRO shifted start point to honor bounds (11 variables).

Problem Characteristics

-----  
 Objective goal: Minimize  
 Number of variables: 11  
   bounded below: 0  
   bounded above: 0  
   bounded below and above: 11  
   fixed: 0  
   free: 0  
 Number of constraints: 10  
   linear equalities: 0  
   nonlinear equalities: 2  
   linear inequalities: 1  
   nonlinear inequalities: 7  
   range: 0  
 Number of nonzeros in Jacobian: 100  
 Number of nonzeros in Hessian: 66

Iter	fCount	Objective	FeasError	OptError	Step	CGits
0	1	8.448589e-008	2.673e-002			
1	2	9.324173e-008	4.664e-003	7.279e+001	1.374e-005	0
2	3	9.045430e-008	2.989e-003	7.634e+001	4.750e-006	0
3	5	8.512982e-008	3.373e-004	8.395e+001	9.369e-006	0
4	13	7.805229e-008	3.302e-004	3.861e+006	9.286e-007	9
5	17	7.681839e-008	3.332e-004	7.996e+000	2.285e-006	0
6	20	7.553900e-008	1.191e-003	2.076e+001	4.192e-006	0
7	27	8.108706e-008	1.146e-003	3.190e-006	7.103e-006	8
8	32	5.114256e-008	1.201e-003	3.053e-006	7.327e-006	2
9	34	5.105849e-008	2.934e-004	1.746e+001	5.261e-006	0

10	37	4.815037e-008	3.268e-004	2.349e+001	5.932e-006	0
11	45	4.867564e-008	1.680e-004	2.958e-006	2.324e-006	9
12	51	4.967880e-008	1.142e-004	1.622e-008	1.641e-006	4
13	57	4.703498e-008	1.141e-004	1.680e-008	2.388e-006	2
14	59	3.509468e-008	1.261e-004	9.745e+000	2.454e-005	0
15	64	7.499798e-008	1.705e-004	4.822e-002	1.613e-005	7
16	68	6.715860e-008	1.446e-004	1.306e+001	4.392e-006	0
17	73	8.792568e-008	9.975e-005	2.890e-002	4.155e-006	3
18	75	8.735542e-008	7.300e-005	1.854e+000	3.749e-006	0
19	81	7.569137e-008	4.433e-005	7.813e-003	2.975e-006	9
20	86	7.788816e-008	1.489e-005	5.369e-003	7.833e-007	2
21	91	8.140603e-008	7.462e-006	1.318e-008	9.677e-007	1
22	97	8.476467e-008	5.856e-006	1.333e-008	8.534e-007	2
23	103	8.929893e-008	4.731e-006	1.349e-008	1.144e-006	1
24	108	9.173277e-008	4.103e-006	1.354e-008	6.116e-007	2
25	113	9.444520e-008	3.483e-006	1.356e-008	6.789e-007	1
26	119	9.275714e-008	4.398e-006	1.366e-008	5.781e-007	2
27	125	9.151105e-008	3.158e-006	1.373e-008	6.205e-007	2
28	132	7.179801e-008	2.944e-005	1.409e-008	3.177e-006	5
29	146	6.194402e-008	8.286e-006	1.428e-008	3.157e-006	3
30	152	6.199447e-008	1.005e-005	2.097e-001	1.062e-006	0
31	159	6.390634e-008	6.392e-007	1.423e-008	1.492e-006	2

EXIT: Locally optimal solution found.

#### Final Statistics

```

-----
Final objective value          = 6.39063404730904e-008
Final feasibility error (abs / rel) = 6.39e-007 / 6.39e-007
Final optimality error (abs / rel) = 1.42e-008 / 1.42e-010
# of iterations                =      31
# of CG iterations             =      76
# of function evaluations      =     159
# of gradient evaluations      =      32
Total program time (secs)     =    0.356 ( 0.361 CPU time)
Time spent in evaluations (secs) =    0.240

```

=====

\* Arquivo de simulação no SMASH (baseado em ideia de Victor Ariel-UFBA)  
 \* Unidades: metro, Volt, Ampere, Farad, Ohm  
 \* Integrador de Seevinck  
 \* 05/06/2011 22:11:49

M1A DD G1A S1A SS N W=6.104084e-005 L=3.312098e-007  
 M2A DD G1A S2A SS N W=6.104084e-005 L=3.312098e-007  
 M3A G1A G3A SS SS N W=7.125741e-005 L=3.312098e-007  
 M4A D4A G4A SS SS N W=7.492709e-005 L=3.312098e-007  
 M6A S2A G4A SS SS N W=4.020534e-005 L=3.312098e-007  
 M7A D7A D7A SS SS N W=6.632586e-005 L=3.312098e-007  
 M8A S1A D7A SS SS N W=6.632586e-005 L=3.312098e-007  
 M10A G3A G3A S1A SS N W=3.823635e-005 L=3.312098e-007  
 M11A G4A G4A S2B SS N W=4.020534e-005 L=3.312098e-007

CINTA S2B SS 6.714708e-011

M1B DD G1B S1B SS N W=6.104084e-005 L=3.312098e-007  
M2B DD G1B S2B SS N W=6.104084e-005 L=3.312098e-007  
M3B G1B G3B SS SS N W=7.125741e-005 L=3.312098e-007  
M4B D4B G4B SS SS N W=7.492709e-005 L=3.312098e-007  
M6B S2B G4B SS SS N W=4.020534e-005 L=3.312098e-007  
M7B D7B D7B SS SS N W=6.632586e-005 L=3.312098e-007  
M8B S1B D7B SS SS N W=6.632586e-005 L=3.312098e-007  
M10B G3B G3B S1B SS N W=3.823635e-005 L=3.312098e-007  
M11B G4B G4B S2A SS N W=4.020534e-005 L=3.312098e-007  
CINTB S2A SS 6.714708e-011

VDD DD 0 DC 6.000000e-001V  
VSS SS 0 DC -6.000000e-001V

.PARAM frac=1  
IIOA DD G1A DC 1.436566e-006  
IO1A DD G3A DC 1.596982e-007  
IO2A DD G4A DC 1.596982e-007  
IBIASA DD D7A 5.574975e-006  
IINA DD D7A AC 1 0  
VOUTA D4A 0 DC 0

IIOB DD G1B DC 1.436566e-006  
IO1B DD G3B DC 1.596982e-007  
IO2B DD G4B DC 1.596982e-007  
IBIASB DD D7B 5.574975e-006  
IINB D7B DD AC 1 0  
VOUTB D4B 0 DC 0

.AC DEC 100 100 100MEG  
.TRAN 100ns 2000us 1000us  
.Trace AC {IOUTDB = DB((I(VOUTB) - I(VOUTA))/2)}  
.Trace TRAN {IOUT = I(VOUTB) - I(VOUTA)}  
.Trace TRAN I(IINA1) I(IINA2)  
.CAPAMIN 1e-18 fF

\* Dados da Tecnologia  
\* BSIM3v3.1 parameters for IBM 0.13  
\* Technology: SCN013  
\*RUN: T93V (#2\_8RF\_8LM\_DM)  
\*SPICE 3f5 Level 8, Star-HSPICE Level 49, UTMOST Level 8  
.MODEL N NMOS ( LEVEL = 8  
+VERSION = 3.1 TNOM = 27 TOX = 3.2E-9  
+XJ = 1E-7 NCH = 2.3549E17 VTH0 = 0.0423138  
+K1 = 0.3322485 K2 = -0.0298343 K3 = 1E-3  
+K3B = 3.6987372 W0 = 1E-7 NLX = 1E-6  
+DVT0W = 0 DVT1W = 0 DVT2W = 0  
+DVT0 = 1.2848863 DVT1 = 0.1532141 DVT2 = 0.2740377  
+U0 = 446.4279855 UA = -1.40947E-10 UB = 3.152185E-18  
+UC = 4.34007E-10 VSAT = 7.64383E4 A0 = 0.1112707  
+AGS = 0.0734135 B0 = 3.913953E-6 B1 = 5E-6  
+KETA = 0.05 A1 = 1.042383E-3 A2 = 0.7518921  
+RDSW = 150 PRWG = 0.3496099 PRWB = 0.1112769

---

```

+WR = 1          WINT = 2.378741E-10 LINT = 1.040187E-8
+DWG = 1.464878E-8 DWB = 9.306917E-9 VOFF = -0.0240657
+NFACTOR = 2.5    CIT = 0          CDSC = 2.4E-4
+CDSCD = 0        CDSCB = 0        ETA0 = 2.750169E-6
+ETAB = -0.0151402 DSUB = 4.056696E-6 PCLM = 1.9784063
+PDIBLC1 = 0.9657497 PDIBLC2 = 0.01 PDIBLCB = 0.1
+DROUT = 0.9990298 PSCBE1 = 7.957109E10 PSCBE2 = 5.015981E-10
+PVAG = 0.535376 DELTA = 0.01     RSH = 6.4
+MOBMOD = 1       PRT = 0          UTE = -1.5
+KT1 = -0.11      KT1L = 0         KT2 = 0.022
+UA1 = 4.31E-9    UB1 = -7.61E-18 UC1 = -5.6E-11
+AT = 3.3E4       WL = 0          WLN = 1
+WW = 0           WWN = 1         WWL = 0
+LL = 0           LLN = 1         LW = 0
+LWN = 1          LWL = 0         CAPMOD = 2
+XPART = 0.5      CGDO = 3.74E-10 CGSO = 3.74E-10
+CGBO = 1E-12     CJ = 9.581155E-4 PB = 0.9756037
+MJ = 0.4044076   CJSW = 1E-10 PPSW = 0.8002027
+MJSW = 0.6       CJSWG = 3.3E-10 PPSWG = 0.8002027
+MJSWG = 0.6      CF = 0          PVTH0 = -7.55492E-4
+PRDSW = 0        PK2 = 1.510875E-3 WKETA = 0.0128032
+LKETA = 0.0141455 PU0 = -3.9867632 PUA = -3.10294E-11
+PUB = 0          PVSAT = 653.2294237 PETA0 = 1E-4
+PKETA = -9.928283E-3 SAREF = 5.5E-7 SBREF = 5.5E-7
+WLOD = 2E-6      KU0 = -4E-6     KVSAT = 0.2
+KVTH0 = 2E-8    LLODKU0 = 1.0867072 STIMOD = 2
+WLODKU0 = 1.0990864 LLODVTH = 1 WLODVTH = 1
+LKU0 = 1E-6     WKU0 = 1E-6     LODETA0 = 1
+LKVTH0 = 1.1E-6 WKVTH0 = 1.1E-6 PKVTH0 = 0
+STK2 = 0        LODK2 = 1       STETA0 = 0

```

```

.MODEL P PMOS (          LEVEL = 8)
+VERSION = 3.1          TNOM = 27          TOX = 3.2E-9
+XJ = 1E-7             NCH = 4.1589E17 VTH0 = -0.2176929
+K1 = 0.2566012       K2 = 6.227669E-3 K3 = 9.8203065
+K3B = 20             W0 = 1E-6          NLX = 2.875116E-7
+DVT0W = 0            DVT1W = 0          DVT2W = 0
+DVT0 = 0.1725649     DVT1 = 0.2214064 DVT2 = 0.1
+U0 = 105.8627093     UA = 1.102916E-9 UB = 3.61005E-21
+UC = -4.6651E-11     VSAT = 7.368516E4 A0 = 2
+AGS = 0.7106176     B0 = 4.50136E-6 B1 = 5E-6
+KETA = 0.029826     A1 = 0.1616816 A2 = 0.3
+RDSW = 106.225597   PRWG = -0.4457476 PRWB = 0.5
+WR = 1              WINT = 0          LINT = 1.483182E-8
+DWG = 9.329926E-10 DWB = -3.202354E-8 VOFF = -0.1022829
+NFACTOR = 1.5332272 CIT = 0          CDSC = 2.4E-4
+CDSCD = 0           CDSCB = 0        ETA0 = 0.0115131
+ETAB = -6.181367E-3 DSUB = 3.781207E-3 PCLM = 0.8846898
+PDIBLC1 = 8.605446E-4 PDIBLC2 = 2.244662E-13 PDIBLCB = 0.1
+DROUT = 0           PSCBE1 = 3.388133E9 PSCBE2 = 1.039776E-9
+PVAG = 0            DELTA = 0.01     RSH = 6.4
+MOBMOD = 1         PRT = 0          UTE = -1.5
+KT1 = -0.11        KT1L = 0         KT2 = 0.022
+UA1 = 4.31E-9      UB1 = -7.61E-18 UC1 = -5.6E-11
+AT = 3.3E4         WL = 0          WLN = 1

```

---

```

+WW = 0      WWN = 1      WWL = 0
+LL = 0      LLN = 1      LW = 0
+LWN = 1     LWL = 0      CAPMOD = 2
+XPART = 0.5  CGDO = 3.42E-10  CGSO = 3.42E-10
+CGBO = 1E-12  CJ = 1.155922E-3  PB = 0.8
+MJ = 0.4423239  CJSW = 1.107787E-10  PBSW = 0.8
+MJSWG = 0.1165941  CJSWG = 4.22E-10  PBSWG = 0.8
+MJSWG = 0.1165941  CF = 0      PVTH0 = 4.037191E-4
+PRDSW = 58.2542288  PK2 = 2.351412E-3  WKETA = 0.0350269
+LKETA = 0.0229421  PU0 = -0.8      PUA = -4.64248E-11
+PUB = 0      PVSAT = 50      PETA0 = 1E-4
+PKETA = -0.0142482  )

```

```

.OP EPS_V=1u VMIN=0 VMAX=1.5 DELTAV=150m EPS_I=100p MAXITER=500
HEURISTICS=-4 TRYALL=YES TIMEOUT=-1
.OPS 1u 100m 1n
.H 1fs 1fs 100ns 125m 2
.METHOD GEAR

```

-----  
Verificação dos resultados finais:  
Verifica restrições de igualdades não-lineares ceq(1) e ceq(2)

ceq(1):  
 $W3/W10 = 1.8636036440273842$   
=  
 $W4/W11 = 1.8636103712646266$

ceq(2):  
 $(1/C * a * \text{phit}) * (I + I_0 - W3/W10 * I_0) = 628382.4548930961200000$   
=  
Kint = 628318.5307179585800000

Verificação da desigualdade linear  
Desigualdade linear:  
Ibias = 0.0000055749748426  
>=  
-iinmin/2 = 0.0000001000000000

Verificação das desigualdades não-lineares c(1) a c(7)  
c(1):  
 $I/I_0 = 7.9955072326860996$   
<=  
 $(W3/W10) * \exp(4/a) - 1 = 86.8239505355002880$

c(2):  
 $-(x(3) + (1/2) * \text{iinmin}) * x(4) * ((x(1) + x(2)) * x(10) / (x(2) * x(6)) - 1) / x(5) + (x(1) + x(2)) * x(4) * x(10) / (x(5) * x(6)) = -0.0000001095058676$   
>=  
 $+bb * \text{Isq} * \exp((4 * \text{phit} - Vto) / (n * aa * \text{phit})) = 0.0000000015510844$

c(3):  
 $W1/L = 184.2965960935744600$   
>=

---

$(1.667*((1-W3*lo/(W10*(l+lo)))*(lbias+(1/2)*iinmax)-lo))/lsq= 20.4711510816731280$

c(4):  
 W3/L= 215.1428026194771200  
 >=  
 $(1.667*(l+lo))/lsq= 6.7767171808521063$

c(5):  
 W7/L= 200.2532975313481500  
 >=  
 $(1.667*(lbias+(1/2)*iinmax))/lsq= 26.7705672719784400$

c(6):  
 W10/L= 115.4445062977757200  
 >=  
 $1.667*lo/lsq= 0.7533446425598113$

c(7):  
 W11/L= 121.3893376065348700  
 >=  
 $1.667*lo/lsq= 0.7533446425598113$

Verifica os limites (bounds) das variáveis:

0.00000000000010000 <=l  
 0.0000012768682589 = l  
 0.0000100000000000 >=l

0.00000000000010000 <=lo  
 0.0000001596982182 = lo  
 0.0000100000000000 >=lo

0.00000000000010000 <=lbias  
 0.0000055749748426 = lbias  
 0.0000100000000000 >=lbias

0.0000002600000000 <=L  
 0.0000003312098159 = L  
 0.0001000000000000 >=L

0.0000002600000000 <= W1=W2  
 0.0000610408416670 = W1=W2  
 0.0001000000000000 >= W1=W2

0.0000002600000000 <= W3  
 0.0000712574080523 = W3  
 0.0001000000000000 >= W3

0.0000002600000000 <= W4  
 0.0000749270889088 = W4  
 0.0001000000000000 >= W4

0.0000002600000000 <=W6=W11  
 0.0000402053401634 = W6=W11  
 0.0001000000000000 >=W6=W11

---

```
0.0000002600000000 <=W7=W8
0.0000663258578129 = W7=W8
0.0001000000000000 >=W7=W8
```

```
0.0000002600000000 <= W10
0.0000382363536800 = W10
0.0001000000000000 >= W10
```

```
0.0000000000100000 <=C
0.0000000000671471 = C
0.0000000001000000 >=C
```

O arquivo seev.cir foi gravado com os dados para entrada no programa SMASH v5

Grato por usar este programa.  
Ricardo Silva

---



---

## EXEMPLO B

OTIMIZAÇÃO DE CIRCUITOS INTEGRADOS - MATLAB OPTIMIZATION  
TOOLBOX - KNITRO

MÉTODOS DOS PONTOS INTERIORES

Digite o valor de "Kint" em radianos por segundo:

? 2\*pi\*10^5

Digite o valor de "iinmax" em Amperes:

? 2e-8

Digite o valor de "iinmin" em Amperes:

? -2e-8

Defina a quantidade máxima de iterações do KNITRO:

? 100

Os valores "default" das tolerâncias são:

TolX= 1e-15 (variáveis)

TolFun=1e-06 (função objetivo)

TolCon=1e-06 (região factível das restrições)

Escolha se quer mudar as tolerâncias.

Digite "S" ou "N": N

---



---

Cálculo da Otimização com o MATLAB Optimization Toolbox/KNITRO  
Método da Otimização: Pontos Interiores

Dados de entrada:

Kint=628318.5307 radianos por segundo

iinmax=2e-008 Amperes

iinmin=-2e-008 Amperes

Valores iniciais das variáveis:

l=2.26000e-007 lo=2.26000e-007 lbias=5.00000e-007  
 L=5.00000e-007 W1=W2=1.00000e-005 W3=1.00000e-005  
 W4=1.00000e-005 W6=W11=5.00000e-006 W7=W8=5.00000e-006  
 W10=1.00000e-005 C=2.00000e-011

Limites ("bounds") inferiores e superiores:

(Unidades: Amperes, metros e Farads)

1.000e-012 <= l <= 1.000e-005  
 1.000e-012 <= lo <= 1.000e-005  
 1.000e-012 <= lbias <= 1.000e-005  
 2.600e-007 <= L <= 1.000e-004  
 2.600e-007 <= W1=W2 <= 1.000e-004  
 2.600e-007 <= W3 <= 1.000e-004  
 2.600e-007 <= W4 <= 1.000e-004  
 2.600e-007 <= W6=W11 <= 1.000e-004  
 2.600e-007 <= W7=W8 <= 1.000e-004  
 2.600e-007 <= W10 <= 1.000e-004  
 1.000e-011 <= C <= 1.000e-010

Tolerâncias especificadas:

TolX= 1e-015 (variáveis)  
 TolFun=1e-006 (função objetivo)  
 TolCon=1e-006 (região factível das restrições)

=====  
 Resultado das iterações:

=====  
 Commercial Ziena License  
 KNITRO 7.0.0  
 Ziena Optimization, Inc.  
 =====

algorithm: 1  
 hessopt: 2  
 honorbnds: 1  
 maxit: 100  
 outlev: 4  
 KNITRO changing bar\_murule from AUTO to 4.  
 KNITRO changing bar\_initpt from AUTO to 2.  
 KNITRO changing bar\_penaltyrule from AUTO to 2.  
 KNITRO changing bar\_penaltycons from AUTO to 1.  
 KNITRO changing linsolver from AUTO to 5.  
 KNITRO shifted start point to honor bounds (11 variables).

Problem Characteristics

-----

Objective goal: Minimize

Number of variables: 11  
 bounded below: 0  
 bounded above: 0  
 bounded below and above: 11  
 fixed: 0  
 free: 0  
 Number of constraints: 10  
 linear equalities: 0  
 nonlinear equalities: 2  
 linear inequalities: 1  
 nonlinear inequalities: 7  
 range: 0  
 Number of nonzeros in Jacobian: 100  
 Number of nonzeros in Hessian: 66

Iter	fCount	Objective	FeasError	OptError	Step	CGits
0	1	8.448589e-008	2.673e-002			
1	2	9.324213e-008	4.665e-003	7.278e+001	1.374e-005	0
2	3	9.045519e-008	2.998e-003	7.631e+001	4.746e-006	0
3	5	8.518068e-008	3.374e-004	8.391e+001	9.303e-006	0
4	13	7.831970e-008	3.303e-004	3.673e-006	9.257e-007	9
5	17	7.710465e-008	3.262e-004	7.820e+000	2.242e-006	0
6	24	7.764033e-008	2.752e-004	6.143e-008	8.029e-007	5
7	32	7.766353e-008	2.627e-004	1.256e-003	9.790e-007	8
8	38	7.497995e-008	2.508e-004	9.914e-004	6.633e-007	3
9	39	5.097929e-008	5.477e-004	3.151e+001	2.130e-005	0
10	41	2.171140e-008	5.846e-004	1.017e+002	2.696e-005	0
11	42	2.851698e-008	1.127e-004	6.569e+001	2.653e-006	0
12	46	3.033186e-008	1.045e-004	5.586e+001	9.430e-007	0
13	49	3.337679e-008	2.406e-005	4.442e+001	1.880e-006	0
14	54	3.704595e-008	4.903e-005	4.703e-004	1.028e-006	1
15	61	3.892781e-008	1.977e-005	1.343e-008	5.153e-007	3
16	67	4.074288e-008	1.753e-005	1.357e-008	4.812e-007	2
17	73	4.241401e-008	1.461e-005	1.368e-008	4.385e-007	2
18	79	4.391451e-008	1.001e-005	1.376e-008	3.936e-007	2
19	85	4.532359e-008	1.115e-005	1.383e-008	3.617e-007	2
20	90	4.682754e-008	9.814e-006	1.389e-008	3.870e-007	1
21	96	4.820582e-008	8.273e-006	1.393e-008	3.529e-007	2
22	102	4.946136e-008	6.804e-006	1.397e-008	3.204e-007	2
23	108	5.059945e-008	5.542e-006	1.399e-008	2.899e-007	2
24	116	5.062260e-008	5.545e-006	1.305e-001	1.168e-007	0
25	122	5.167162e-008	2.753e-006	1.402e-008	2.749e-007	3

26	131	5.211041e-008	2.732e-006	1.402e-008	1.119e-007	11
27	139	5.243053e-008	2.713e-006	1.403e-008	8.241e-008	9
28	147	5.262984e-008	2.696e-006	1.403e-008	5.125e-008	8
29	153	5.274599e-008	2.687e-006	1.403e-008	2.872e-008	4
30	158	5.293026e-008	2.671e-006	1.403e-008	4.746e-008	1
31	163	5.311586e-008	2.655e-006	1.403e-008	4.764e-008	1
32	168	5.329637e-008	2.639e-006	1.403e-008	4.729e-008	1
33	174	5.387074e-008	2.611e-006	1.404e-008	1.496e-007	2
34	179	5.443575e-008	2.579e-006	1.404e-008	1.496e-007	1
35	192	5.623565e-008	2.685e-006	1.405e-008	4.755e-007	2
36	202	5.623551e-008	2.681e-006	4.009e-002	2.939e-008	0
37	207	5.682705e-008	2.642e-006	1.405e-008	1.551e-007	1
38	213	5.867078e-008	2.718e-006	1.405e-008	4.841e-007	2
39	221	5.995666e-008	2.684e-006	1.404e-008	3.347e-007	4
40	226	6.127318e-008	2.640e-006	1.403e-008	3.410e-007	1
41	231	6.260838e-008	2.579e-006	1.402e-008	3.449e-007	1
42	236	6.396552e-008	2.502e-006	1.400e-008	3.489e-007	1
43	241	6.533971e-008	2.404e-006	1.398e-008	3.517e-007	1
44	246	6.672343e-008	2.279e-006	1.396e-008	3.536e-007	1
45	251	6.812415e-008	2.127e-006	1.393e-008	3.560e-007	1
46	256	6.953365e-008	1.940e-006	1.391e-008	3.570e-007	1
47	266	6.950846e-008	1.936e-006	6.540e-002	3.090e-008	0
48	271	7.018378e-008	1.792e-006	1.390e-008	1.714e-007	1
49	277	7.049380e-008	1.716e-006	1.389e-008	7.688e-008	2
50	282	7.080106e-008	1.639e-006	1.388e-008	7.625e-008	1
51	287	7.111014e-008	1.561e-006	1.388e-008	7.640e-008	1
52	292	7.142411e-008	1.482e-006	1.387e-008	7.646e-008	1
53	297	7.173646e-008	1.402e-006	1.387e-008	7.614e-008	1
54	302	7.205126e-008	1.321e-006	1.386e-008	7.625e-008	1
55	307	7.236516e-008	1.239e-006	1.386e-008	7.613e-008	1
56	312	7.268041e-008	1.156e-006	1.385e-008	7.618e-008	1
57	317	7.299837e-008	1.072e-006	1.385e-008	7.622e-008	1
58	326	7.294858e-008	2.379e-006	1.188e-001	6.164e-008	0
59	332	7.365262e-008	1.834e-006	1.384e-008	1.664e-007	2
60	334	7.351128e-008	1.524e-006	2.467e-001	6.281e-008	0
61	340	7.413891e-008	1.574e-006	1.383e-008	1.487e-007	2
62	342	7.410681e-008	9.895e-007	8.429e-002	5.247e-008	0
63	349	7.436231e-008	6.634e-007	1.383e-008	6.031e-008	3

EXIT: Locally optimal solution found.

#### Final Statistics

-----

Final objective value = 7.43623076867633e-008

Final feasibility error (abs / rel) = 6.63e-007 / 6.63e-007

Final optimality error (abs / rel) = 1.38e-008 / 1.38e-010

```

# of iterations           =      63
# of CG iterations       =      119
# of function evaluations =     349
# of gradient evaluations =      64
Total program time (secs) =    0.592 ( 0.571 CPU time)
Time spent in evaluations (secs) =    0.399

```

```
=====
```

```

* Arquivo de simulação no SMASH (baseado em ideia de Victor Ariel-UFBA)
* Unidades: metro, Volt, Ampere, Farad, Ohm
* Integrador de Seevinck
* 05/06/2011 22:07:44

```

```

M1A DD G1A S1A SS N W=5.665418e-005 L=3.621820e-007
M2A DD G1A S2A SS N W=5.665418e-005 L=3.621820e-007
M3A G1A G3A SS SS N W=7.118679e-005 L=3.621820e-007
M4A D4A G4A SS SS N W=6.842477e-005 L=3.621820e-007
M6A S2A G4A SS SS N W=3.394900e-005 L=3.621820e-007
M7A D7A D7A SS SS N W=6.218026e-005 L=3.621820e-007
M8A S1A D7A SS SS N W=6.218026e-005 L=3.621820e-007
M10A G3A G3A S1A SS N W=3.531681e-005 L=3.621820e-007
M11A G4A G4A S2B SS N W=3.394900e-005 L=3.621820e-007
CINTA S2B SS 7.818866e-011

```

```

M1B DD G1B S1B SS N W=5.665418e-005 L=3.621820e-007
M2B DD G1B S2B SS N W=5.665418e-005 L=3.621820e-007
M3B G1B G3B SS SS N W=7.118679e-005 L=3.621820e-007
M4B D4B G4B SS SS N W=6.842477e-005 L=3.621820e-007
M6B S2B G4B SS SS N W=3.394900e-005 L=3.621820e-007
M7B D7B D7B SS SS N W=6.218026e-005 L=3.621820e-007
M8B S1B D7B SS SS N W=6.218026e-005 L=3.621820e-007
M10B G3B G3B S1B SS N W=3.531681e-005 L=3.621820e-007
M11B G4B G4B S2A SS N W=3.394900e-005 L=3.621820e-007
CINTB S2A SS 7.818866e-011

```

```

VDD DD 0 DC 6.000000e-001V
VSS SS 0 DC -6.000000e-001V

```

```

.PARAM frac=1
II0A DD G1A DC 2.427589e-006
I01A DD G3A DC 5.465256e-007
I02A DD G4A DC 5.465256e-007
IBIASA DD D7A 3.204018e-006
IINA DD D7A AC 1 0
VOUTA D4A 0 DC 0

```

```

IIOB DD G1B DC 2.427589e-006
I01B DD G3B DC 5.465256e-007
I02B DD G4B DC 5.465256e-007
IBIASB DD D7B 3.204018e-006
IINB D7B DD AC 1 0
VOUTB D4B 0 DC 0

```

```

.AC DEC 100 100 100MEG
.TRAN 100ns 2000us 1000us
.Trace AC {IOUTDB = DB((I(VOUTB) - I(VOUTA))/2)}
.Trace TRAN {IOUT = I(VOUTB) - I(VOUTA)}
.Trace TRAN I(IINA1) I(IINA2)
.CAPAMIN 1e-18 fF

```

\* Dados da Tecnologia

\* BSIM3v3.1 parameters for IBM 0.13

\* Technology: SCN013

\*RUN: T93V (#2\_8RF\_8LM\_DM)

\*SPICE 3f5 Level 8, Star-HSPICE Level 49, UTMOST Level 8

```

.MODEL N NMOS (          LEVEL = 8
+VERSION = 3.1          TNOM = 27          TOX = 3.2E-9
+XJ = 1E-7              NCH = 2.3549E17    VTH0 = 0.0423138
+K1 = 0.3322485        K2 = -0.0298343    K3 = 1E-3
+K3B = 3.6987372      W0 = 1E-7          NLX = 1E-6
+DVT0W = 0             DVT1W = 0          DVT2W = 0
+DVT0 = 1.2848863     DVT1 = 0.1532141    DVT2 = 0.2740377
+U0 = 446.4279855     UA = -1.40947E-10  UB = 3.152185E-18
+UC = 4.34007E-10     VSAT = 7.64383E4    A0 = 0.1112707
+AGS = 0.0734135     B0 = 3.913953E-6    B1 = 5E-6
+KETA = 0.05          A1 = 1.042383E-3    A2 = 0.7518921
+RDSW = 150           PRWG = 0.3496099    PRWB = 0.1112769
+WR = 1               WINT = 2.378741E-10 LINT = 1.040187E-8
+DWG = 1.464878E-8    DWB = 9.306917E-9   VOFF = -0.0240657
+NFACTOR = 2.5        CIT = 0              CDSC = 2.4E-4
+CDSCD = 0            CDSCB = 0            ETA0 = 2.750169E-6
+ETAB = -0.0151402    DSUB = 4.056696E-6  PCLM = 1.9784063
+PDIBLC1 = 0.9657497 PDIBLC2 = 0.01       PDIBLCB = 0.1
+DROUT = 0.9990298    PSCBE1 = 7.957109E10 PSCBE2 = 5.015981E-
10
+PVAG = 0.535376      DELTA = 0.01         RSH = 6.4
+MOBMOD = 1           PRT = 0              UTE = -1.5
+KT1 = -0.11          KT1L = 0             KT2 = 0.022
+UA1 = 4.31E-9         UB1 = -7.61E-18     UC1 = -5.6E-11
+AT = 3.3E4           WL = 0               WLN = 1
+WW = 0               WWN = 1              WWL = 0

```

```

+LL = 0      LLN = 1      LW = 0
+LWN = 1     LWL = 0     CAPMOD = 2
+XPART = 0.5 CGDO = 3.74E-10 CGSO = 3.74E-10
+CGBO = 1E-12 CJ = 9.581155E-4 PB = 0.9756037
+MJ = 0.4044076 CJSW = 1E-10 PBSW = 0.8002027
+MJSW = 0.6    CJSWG = 3.3E-10 PBSWG = 0.8002027
+MJSWG = 0.6   CF = 0     PVTH0 = -7.55492E-4
+PRDSW = 0     PK2 = 1.510875E-3 WKETA = 0.0128032
+LKETA = 0.0141455 PU0 = -3.9867632 PUA = -3.10294E-11
+PUB = 0      PVSAT = 653.2294237 PETA0 = 1E-4
+PKETA = -9.928283E-3 SAREF = 5.5E-7 SBREF = 5.5E-7
+WLOD = 2E-6   KU0 = -4E-6   KVSAT = 0.2
+KVTH0 = 2E-8 LLODKU0 = 1.0867072 STIMOD = 2
+WLODKU0 = 1.0990864 LLODVTH = 1 WLODVTH = 1
+LKU0 = 1E-6   WKU0 = 1E-6   LODETA0 = 1
+LKVTH0 = 1.1E-6 WKVTH0 = 1.1E-6 PKVTH0 = 0
+STK2 = 0     LODK2 = 1     STETA0 = 0

```

```

.MODEL P PMOS (          LEVEL = 8)
+VERSION = 3.1          TNOM = 27          TOX = 3.2E-9
+XJ = 1E-7             NCH = 4.1589E17      VTH0 = -0.2176929
+K1 = 0.2566012       K2 = 6.227669E-3      K3 = 9.8203065
+K3B = 20             W0 = 1E-6          NLX = 2.875116E-7
+DVT0W = 0           DVT1W = 0          DVT2W = 0
+DVT0 = 0.1725649    DVT1 = 0.2214064    DVT2 = 0.1
+U0 = 105.8627093    UA = 1.102916E-9    UB = 3.61005E-21
+UC = -4.6651E-11    VSAT = 7.368516E4   A0 = 2
+AGS = 0.7106176     B0 = 4.50136E-6     B1 = 5E-6
+KETA = 0.029826     A1 = 0.1616816     A2 = 0.3
+RDSW = 106.225597   PRWG = -0.4457476   PRWB = 0.5
+WR = 1             WINT = 0          LINT = 1.483182E-8
+DWG = 9.329926E-10 DWB = -3.202354E-8 VOFF = -0.1022829
+NFACTOR = 1.5332272 CIT = 0          CDSC = 2.4E-4
+CDSCD = 0          CDSCB = 0          ETA0 = 0.0115131
+ETAB = -6.181367E-3 DSUB = 3.781207E-3 PCLM = 0.8846898
+PDIBLC1 = 8.605446E-4 PDIBLC2 = 2.244662E-13 PDIBLCB = 0.1
+DROUT = 0          PSCBE1 = 3.388133E9   PSCBE2 = 1.039776E-9
+PVAG = 0           DELTA = 0.01         RSH = 6.4
+MOBMOD = 1         PRT = 0            UTE = -1.5
+KT1 = -0.11        KT1L = 0           KT2 = 0.022
+UA1 = 4.31E-9       UB1 = -7.61E-18    UC1 = -5.6E-11
+AT = 3.3E4          WL = 0            WLN = 1
+WW = 0             WWN = 1           WWL = 0
+LL = 0            LLN = 1           LW = 0
+LWN = 1           LWL = 0           CAPMOD = 2
+XPART = 0.5       CGDO = 3.42E-10    CGSO = 3.42E-10

```

```

+CGBO = 1E-12      CJ = 1.155922E-3  PB = 0.8
+MJ = 0.4423239   CJSW = 1.107787E-10  PBSW = 0.8
+MJSW = 0.1165941  CJSWG = 4.22E-10   PBSWG = 0.8
+MJSWG = 0.1165941  CF = 0           PVTH0 = 4.037191E-4
+PRDSW = 58.2542288  PK2 = 2.351412E-3  WKETA = 0.0350269
+LKETA = 0.0229421  PU0 = -0.8       PUA = -4.64248E-11
+PUB = 0           PVSAT = 50         PETA0 = 1E-4
+PKETA = -0.0142482  )

```

```

.OP EPS_V=1u VMIN=0 VMAX=1.5 DELTAV=150m EPS_I=100p
MAXITER=500 HEURISTICS=-4 TRYALL=YES TIMEOUT=-1
.EPS 1u 100m 1n
.H 1fs 1fs 100ns 125m 2
.METHOD GEAR

```

-----  
Verificação dos resultados finais:  
Verifica restrições de igualdades não-lineares ceq(1) e ceq(2)

ceq(1):  
W3/W10= 2.0156633728477851  
=  
W4/W11= 2.0155164188389372

ceq(2):  
(1/C\*a\*phit)\*(I+I0-W3/W10\*I0)= 628257.8902101792600000  
=  
Kint= 628318.5307179585800000

Verificação da desigualdade linear

Desigualdade linear:  
Ibias= 0.0000032040176613  
>=  
-iinmin/2= 0.0000000100000000

Verificação das desigualdades não-lineares c(1) a c(7)

c(1):  
I/I0= 3.4418577330263429  
<=  
(W3/W10)\*exp(4/a)-1= 93.9898981580883510

c(2):  
-(x(3)+(1/2)\*iinmin)\*x(4)\*((x(1)+x(2))\*x(10)/(x(2)\*x(6))-  
1)/x(5)+(x(1)+x(2))\*x(4)\*x(10)/(x(5)\*x(6))=  
-0.0000000168783047  
>=  
+bb\*Isq\*exp((4\*phit-Vto)/(n\*aa\*phit))= 0.0000000015510844

c(3):  
W1/L= 156.4246137245231700  
>=  
 $(1.667 * ((1 - W3 * lo / (W10 * (l + lo))) * (lbias + (1/2) * iinmax) - lo)) / lsq =$   
5.7032564146130236

c(4):  
W3/L= 196.5497705620407200  
>=  
 $(1.667 * (l + lo)) / lsq = 11.4516686430425750$

c(5):  
W7/L= 171.6823461631289800  
>=  
 $(1.667 * (lbias + (1/2) * iinmax)) / lsq = 15.1614902975111700$

c(6):  
W10/L= 97.5112080765498770  
>=  
 $1.667 * lo / lsq = 2.5781259399409633$

c(7):  
W11/L= 93.7346400453981800  
>=  
 $1.667 * lo / lsq = 2.5781259399409633$

Verifica os limites (bounds) das variáveis:

0.0000000000010000 <= l  
0.0000018810632996 = l  
0.0000100000000000 >= l

0.0000000000010000 <= lo  
0.0000005465255817 = lo  
0.0000100000000000 >= lo

0.0000000000010000 <= lbias  
0.0000032040176613 = lbias  
0.0000100000000000 >= lbias

0.0000002600000000 <= L  
0.0000003621820177 = L  
0.0001000000000000 >= L

0.0000002600000000 <= W1=W2

0.0000566541822173 = W1=W2  
 0.0001000000000000 >= W1=W2

0.0000002600000000 <= W3  
 0.0000711867924814 = W3  
 0.0001000000000000 >= W3

0.0000002600000000 <= W4  
 0.0000684247690404 = W4  
 0.0001000000000000 >= W4

0.0000002600000000 <=W6=W11  
 0.0000339490010604 = W6=W11  
 0.0001000000000000 >=W6=W11

0.0000002600000000 <=W7=W8  
 0.0000621802585375 = W7=W8  
 0.0001000000000000 >=W7=W8

0.0000002600000000 <= W10  
 0.0000353168060899 = W10  
 0.0001000000000000 >= W10

0.0000000000100000 <=C  
 0.0000000000781887 = C  
 0.0000000000100000 >=C

O arquivo seev.cir foi gravado com os dados para entrada no programa SMASH v5

Grato por usar este programa.  
 Ricardo Silva

=====

### EXEMPLO C

OTIMIZAÇÃO DE CIRCUITOS INTEGRADOS - MATLAB OPTIMIZATION  
 TOOLBOX - KNITRO

MÉTODOS DOS PONTOS INTERIORES

Digite o valor de "Kint" em radianos por segundo:

? 2\*pi\*10^6

Digite o valor de "iinmax" em Amperes:

? 2e-8

Digite o valor de "iinmin" em Amperes:

? -2e-8

Defina a quantidade máxima de iterações do KNITRO:

? 50

Os valores "default" das tolerâncias são:

TolX= 1e-15 (variáveis)

TolFun=1e-06 (função objetivo)

TolCon=1e-06 (região factível das restrições)

Escolha se quer mudar as tolerâncias.

Digite "S" ou "N": N

=====

Cálculo da Otimização com o MATLAB Opimization Toolbox/KNITRO  
Método da Otimização: Pontos Interiores

Dados de entrada:

Kint=6283185.3072 radianos por segundo

iinmax=2e-008 Amperes

iinmin=-2e-008 Amperes

Valores iniciais das variáveis:

I=2.26000e-007 Io=2.26000e-007 Ibias=5.00000e-007

L=5.00000e-007 W1=W2=1.00000e-005 W3=1.00000e-005

W4=1.00000e-005 W6=W11=5.00000e-006 W7=W8=5.00000e-006

W10=1.00000e-005 C=2.00000e-011

Limites ("bounds") inferiores e superiores:

(Unidades: Amperes, metros e Farads)

1.000e-012 <= I <= 1.000e-005

1.000e-012 <= Io <= 1.000e-005

1.000e-012 <= Ibias <= 1.000e-005

2.600e-007 <= L <= 1.000e-004

2.600e-007 <= W1=W2 <= 1.000e-004

2.600e-007 <= W3 <= 1.000e-004

2.600e-007 <= W4 <= 1.000e-004

2.600e-007 <= W6=W11 <= 1.000e-004

2.600e-007 <= W7=W8 <= 1.000e-004

2.600e-007 <= W10 <= 1.000e-004

1.000e-011 <= C <= 1.000e-010

Tolerâncias especificadas:

TolX= 1e-015 (variáveis)

TolFun=1e-006 (função objetivo)

TolCon=1e-006 (região factível das restrições)

=====

Resultado das iterações:

```

=====
Commercial Ziena License
KNITRO 7.0.0
Ziena Optimization, Inc.
=====

```

```

algorithm:      1
hessopt:       2
honorbnds:     1
maxit:         50
outlev:        4
KNITRO changing bar_murule from AUTO to 4.
KNITRO changing bar_initpt from AUTO to 2.
KNITRO changing bar_penaltyrule from AUTO to 2.
KNITRO changing bar_penaltycons from AUTO to 1.
KNITRO changing linsolver from AUTO to 5.
KNITRO shifted start point to honor bounds (11 variables).

```

#### Problem Characteristics

```

-----
Objective goal: Minimize
Number of variables:      11
  bounded below:          0
  bounded above:          0
  bounded below and above: 11
  fixed:                  0
  free:                   0
Number of constraints:    10
  linear equalities:      0
  nonlinear equalities:   2
  linear inequalities:    1
  nonlinear inequalities:  7
  range:                  0
Number of nonzeros in Jacobian: 100
Number of nonzeros in Hessian:  66

```

Iter	fCount	Objective	FeasError	OptError	Step	CGits
0	1	8.448589e-008	2.982e-002			
1	2	7.568389e-008	1.456e-003	1.510e+002	1.427e-005	0
2	3	7.446947e-008	9.586e-004	1.499e+002	2.355e-006	0
3	6	7.075019e-008	1.504e-003	1.646e+002	3.627e-006	0
4	9	6.697858e-008	2.154e-003	1.871e+002	3.220e-006	0
5	11	5.686200e-008	2.494e-003	3.403e+002	7.344e-006	0
6	12	2.933417e-008	1.170e-003	1.165e+004	1.994e-005	0

7	13	1.085700e-008	3.323e-004	4.516e+004	2.127e-005	0
8	14	1.085595e-008	2.287e-005	4.502e+004	4.220e-007	0
9	15	1.109773e-008	9.946e-006	3.171e+004	1.424e-006	0
10	16	1.119874e-008	4.229e-007	7.582e+002	1.258e-007	0
11	17	3.189732e-008	1.534e-004	1.658e+001	2.005e-005	0
12	18	2.533160e-008	6.559e-004	9.511e+001	1.828e-005	0
13	19	3.721358e-008	1.539e-005	7.633e+001	1.249e-005	0
14	21	3.915930e-008	1.145e-005	1.743e+001	2.736e-006	0
15	23	3.888888e-008	2.088e-006	1.234e+001	1.562e-006	0
16	25	3.886268e-008	1.190e-006	1.185e+001	8.379e-007	0
17	27	3.886938e-008	4.836e-007	1.199e+001	3.395e-007	0
18	29	3.887324e-008	1.904e-007	1.207e+001	1.358e-007	0
19	31	3.887490e-008	7.655e-008	1.210e+001	5.588e-008	0
20	33	3.887561e-008	3.250e-008	1.212e+001	2.423e-008	0
21	35	3.887595e-008	1.494e-008	1.212e+001	1.133e-008	0
22	37	3.887611e-008	7.594e-009	1.213e+001	5.836e-009	0
23	39	3.887618e-008	2.737e-009	1.213e+001	2.148e-009	0
24	41	3.887620e-008	9.409e-010	1.213e+001	7.815e-010	0
25	43	3.887621e-008	2.783e-010	1.213e+001	2.786e-010	0
26	45	3.887621e-008	3.512e-011	1.213e+001	1.033e-010	0
27	47	3.887621e-008	5.276e-011	1.213e+001	6.040e-011	0
28	54	3.887621e-008	1.002e-018	5.861e-008	1.299e-012	2

EXIT: Locally optimal solution found.

#### Final Statistics

-----

Final objective value = 3.88762086214811e-008  
 Final feasibility error (abs / rel) = 1.00e-018 / 1.00e-018  
 Final optimality error (abs / rel) = 5.86e-008 / 5.86e-010  
 # of iterations = 28  
 # of CG iterations = 2  
 # of function evaluations = 54  
 # of gradient evaluations = 29  
 Total program time (secs) = 0.483 ( 0.481 CPU time)  
 Time spent in evaluations (secs) = 0.401

=====

\* Arquivo de simulação no SMASH (baseado em ideia de Victor Ariel-UFBA)  
 \* Unidades: metro, Volt, Ampere, Farad, Ohm  
 \* Integrador de Seevinck  
 \* 05/06/2011 21:56:29

M1A DD G1A S1A SS N W=5.381529e-005 L=3.322052e-007  
 M2A DD G1A S2A SS N W=5.381529e-005 L=3.322052e-007

M3A G1A G3A SS SS N W=6.638553e-005 L=3.322052e-007  
 M4A D4A G4A SS SS N W=5.932573e-005 L=3.322052e-007  
 M6A S2A G4A SS SS N W=6.094718e-005 L=3.322052e-007  
 M7A D7A D7A SS SS N W=6.868056e-005 L=3.322052e-007  
 M8A S1A D7A SS SS N W=6.868056e-005 L=3.322052e-007  
 M10A G3A G3A S1A SS N W=6.819994e-005 L=3.322052e-007  
 M11A G4A G4A S2B SS N W=6.094718e-005 L=3.322052e-007  
 CINTA S2B SS 4.067521e-011

M1B DD G1B S1B SS N W=5.381529e-005 L=3.322052e-007  
 M2B DD G1B S2B SS N W=5.381529e-005 L=3.322052e-007  
 M3B G1B G3B SS SS N W=6.638553e-005 L=3.322052e-007  
 M4B D4B G4B SS SS N W=5.932573e-005 L=3.322052e-007  
 M6B S2B G4B SS SS N W=6.094718e-005 L=3.322052e-007  
 M7B D7B D7B SS SS N W=6.868056e-005 L=3.322052e-007  
 M8B S1B D7B SS SS N W=6.868056e-005 L=3.322052e-007  
 M10B G3B G3B S1B SS N W=6.819994e-005 L=3.322052e-007  
 M11B G4B G4B S2A SS N W=6.094718e-005 L=3.322052e-007  
 CINTB S2A SS 4.067521e-011

VDD DD 0 DC 6.000000e-001V  
 VSS SS 0 DC -6.000000e-001V

.PARAM frac=1  
 IIOA DD G1A DC 7.531331e-006  
 IO1A DD G3A DC 6.499732e-007  
 IO2A DD G4A DC 6.499732e-007  
 IBIASA DD D7A 7.264849e-006  
 IINA DD D7A AC 1 0  
 VOUTA D4A 0 DC 0

IIOB DD G1B DC 7.531331e-006  
 IO1B DD G3B DC 6.499732e-007  
 IO2B DD G4B DC 6.499732e-007  
 IBIASB DD D7B 7.264849e-006  
 IINB D7B DD AC 1 0  
 VOUTB D4B 0 DC 0

.AC DEC 100 100 100MEG  
 .TRAN 100ns 2000us 1000us  
 .Trace AC {IOUTDB = DB((I(VOUTB) - I(VOUTA))/2)}  
 .Trace TRAN {IOUT = I(VOUTB) - I(VOUTA)}  
 .Trace TRAN I(IINA1) I(IINA2)  
 .CAPAMIN 1e-18 fF

\* Dados da Tecnologia

```

* BSIM3v3.1 parameters for IBM 0.13
* Technology: SCN013
*RUN: T93V (#2_8RF_8LM_DM)
*SPICE 3f5 Level 8, Star-HSPICE Level 49, UTMOST Level 8
.MODEL N NMOS (          LEVEL = 8
+VERSION = 3.1          TNOM = 27          TOX = 3.2E-9
+XJ = 1E-7             NCH = 2.3549E17     VTH0 = 0.0423138
+K1 = 0.3322485       K2 = -0.0298343     K3 = 1E-3
+K3B = 3.6987372      W0 = 1E-7          NLX = 1E-6
+DVT0W = 0            DVT1W = 0          DVT2W = 0
+DVT0 = 1.2848863     DVT1 = 0.1532141     DVT2 = 0.2740377
+U0 = 446.4279855     UA = -1.40947E-10   UB = 3.152185E-18
+UC = 4.34007E-10     VSAT = 7.64383E4    A0 = 0.1112707
+AGS = 0.0734135     B0 = 3.913953E-6    B1 = 5E-6
+KETA = 0.05          A1 = 1.042383E-3     A2 = 0.7518921
+RDSW = 150           PRWG = 0.3496099     PRWB = 0.1112769
+WR = 1               WINT = 2.378741E-10 LINT = 1.040187E-8
+DWG = 1.464878E-8    DWB = 9.306917E-9   VOFF = -0.0240657
+NFACTOR = 2.5        CIT = 0            CDSC = 2.4E-4
+CDSCD = 0            CDSCB = 0            ETA0 = 2.750169E-6
+ETAB = -0.0151402    DSUB = 4.056696E-6  PCLM = 1.9784063
+PDIBLC1 = 0.9657497 PDIBLC2 = 0.01       PDIBLCB = 0.1
+DROUT = 0.9990298    PSCBE1 = 7.957109E10 PSCBE2 = 5.015981E-
10
+PVAG = 0.535376      DELTA = 0.01         RSH = 6.4
+MOBMOD = 1           PRT = 0              UTE = -1.5
+KT1 = -0.11          KT1L = 0             KT2 = 0.022
+UA1 = 4.31E-9         UB1 = -7.61E-18      UC1 = -5.6E-11
+AT = 3.3E4           WL = 0               WLN = 1
+WW = 0               WWN = 1              WWL = 0
+LL = 0               LLN = 1              LW = 0
+LWN = 1              LWL = 0              CAPMOD = 2
+XPART = 0.5          CGDO = 3.74E-10      CGSO = 3.74E-10
+CGBO = 1E-12         CJ = 9.581155E-4     PB = 0.9756037
+MJ = 0.4044076       CJSW = 1E-10         PBSW = 0.8002027
+MJSW = 0.6           CJSWG = 3.3E-10      PBSWG = 0.8002027
+MJSWG = 0.6          CF = 0               PVTH0 = -7.55492E-4
+PRDSW = 0            PK2 = 1.510875E-3    WKETA = 0.0128032
+LKETA = 0.0141455     PU0 = -3.9867632     PUA = -3.10294E-11
+PUB = 0              PVSAT = 653.2294237  PETA0 = 1E-4
+PKETA = -9.928283E-3 SAREF = 5.5E-7       SBREF = 5.5E-7
+WLOD = 2E-6          KU0 = -4E-6          KVSAT = 0.2
+KVTH0 = 2E-8         LLODKU0 = 1.0867072  STIMOD = 2
+WLODKU0 = 1.0990864  LLODVTH = 1          WLODVTH = 1
+LKU0 = 1E-6          WKU0 = 1E-6          LODETA0 = 1
+LKVTH0 = 1.1E-6     WKVTH0 = 1.1E-6     PKVTH0 = 0

```

+STK2 = 0            LODK2 = 1            STETA0 = 0

```
.MODEL P PMOS (                            LEVEL = 8)
+VERSION = 3.1            TNOM = 27            TOX = 3.2E-9
+XJ = 1E-7            NCH = 4.1589E17            VTH0 = -0.2176929
+K1 = 0.2566012            K2 = 6.227669E-3            K3 = 9.8203065
+K3B = 20            W0 = 1E-6            NLX = 2.875116E-7
+DVT0W = 0            DVT1W = 0            DVT2W = 0
+DVT0 = 0.1725649            DVT1 = 0.2214064            DVT2 = 0.1
+U0 = 105.8627093            UA = 1.102916E-9            UB = 3.61005E-21
+UC = -4.6651E-11            VSAT = 7.368516E4            A0 = 2
+AGS = 0.7106176            B0 = 4.50136E-6            B1 = 5E-6
+KETA = 0.029826            A1 = 0.1616816            A2 = 0.3
+RDSW = 106.225597            PRWG = -0.4457476            PRWB = 0.5
+WR = 1            WINT = 0            LINT = 1.483182E-8
+DWG = 9.329926E-10            DWB = -3.202354E-8            VOFF = -0.1022829
+NFACTOR = 1.5332272            CIT = 0            CDSC = 2.4E-4
+CDSCD = 0            CDSCB = 0            ETA0 = 0.0115131
+ETAB = -6.181367E-3            DSUB = 3.781207E-3            PCLM = 0.8846898
+PDIBLC1 = 8.605446E-4            PDIBLC2 = 2.244662E-13            PDIBLCB = 0.1
+DROUT = 0            PSCBE1 = 3.388133E9            PSCBE2 = 1.039776E-9
+PVAG = 0            DELTA = 0.01            RSH = 6.4
+MOBMOD = 1            PRT = 0            UTE = -1.5
+KT1 = -0.11            KT1L = 0            KT2 = 0.022
+UA1 = 4.31E-9            UB1 = -7.61E-18            UC1 = -5.6E-11
+AT = 3.3E4            WL = 0            WLN = 1
+WW = 0            WWN = 1            WWL = 0
+LL = 0            LLN = 1            LW = 0
+LWN = 1            LWL = 0            CAPMOD = 2
+XPART = 0.5            CGDO = 3.42E-10            CGSO = 3.42E-10
+CGBO = 1E-12            CJ = 1.155922E-3            PB = 0.8
+MJ = 0.4423239            CJSW = 1.107787E-10            PBSW = 0.8
+MJSW = 0.1165941            CJSWG = 4.22E-10            PBSWG = 0.8
+MJSWG = 0.1165941            CF = 0            PVTH0 = 4.037191E-4
+PRDSW = 58.2542288            PK2 = 2.351412E-3            WKETA = 0.0350269
+LKETA = 0.0229421            PU0 = -0.8            PUA = -4.64248E-11
+PUB = 0            PVSAT = 50            PETA0 = 1E-4
+PKETA = -0.0142482            )
```

```
.OP EPS_V=1u VMIN=0 VMAX=1.5 DELTAV=150m EPS_I=100p
MAXITER=500 HEURISTICS=-4 TRYALL=YES TIMEOUT=-1
.EPS 1u 100m 1n
.H 1fs 1fs 100ns 125m 2
.METHOD GEAR
```

---

Verificação dos resultados finais:

Verifica restrições de igualdades não-lineares ceq(1) e ceq(2)

ceq(1):

$$W3/W10 = 0.9733957659037996$$

=

$$W4/W11 = 0.9733957659037994$$

ceq(2):

$$(1/C * a * \text{phit}) * (I + I_0 - W3/W10 * I_0) = 6283185.3071795860000000$$

=

$$\text{Kint} = 6283185.3071795860000000$$

Verificação da desigualdade linear

Desigualdade linear:

$$I_{\text{bias}} = 0.0000072648493259$$

>=

$$-i_{\text{inmin}}/2 = 0.0000000100000000$$

Verificação das desigualdades não-lineares c(1) a c(7)

c(1):

$$I/I_0 = 10.5871405953303360$$

<=

$$(W3/W10) * \exp(4/a) - 1 = 44.8721262271499060$$

c(2):

$$-(x(3) + (1/2) * i_{\text{inmin}}) * x(4) * ((x(1) + x(2)) * x(10) / (x(2) * x(6)) - 1) / x(5) + (x(1) + x(2)) * x(4) * x(10) / (x(5) * x(6)) =$$

$$-0.0000004405622353$$

>=

$$+bb * I_{\text{sq}} * \exp((4 * \text{phit} - V_{\text{to}}) / (n * aa * \text{phit})) = 0.0000000015510844$$

c(3):

$$W1/L = 161.9941233861109400$$

>=

$$(1.667 * ((1 - W3 * I_0 / (W10 * (I + I_0))) * (I_{\text{bias}} + (1/2) * i_{\text{inmax}}) - I_0)) / I_{\text{sq}} =$$

$$28.3686299548846000$$

c(4):

$$W3/L = 199.8329133892289100$$

>=

$$(1.667 * (I + I_0)) / I_{\text{sq}} = 35.5275566346998130$$

c(5):  
W7/L= 206.7413761741301000  
>=  
 $(1.667*(lbias+(1/2)*iinmax))/lsq= 34.3176575537220000$

c(6):  
W10/L= 205.2946195052366000  
>=  
 $1.667*lo/lsq= 3.0661194055949892$

c(7):  
W11/L= 183.4624469663164300  
>=  
 $1.667*lo/lsq= 3.0661194055949892$

Verifica os limites (bounds) das variáveis:

0.000000000010000 <=I  
0.0000068813573413 = I  
0.0000100000000000 >=I

0.000000000010000 <=lo  
0.0000006499731707 = lo  
0.0000100000000000 >=lo

0.000000000010000 <=lbias  
0.0000072648493259 = lbias  
0.0000100000000000 >=lbias

0.0000002600000000 <=L  
0.0000003322052078 = L  
0.0001000000000000 >=L

0.0000002600000000 <= W1=W2  
0.0000538152914276 = W1=W2  
0.0001000000000000 >= W1=W2

0.0000002600000000 <= W3  
0.0000663855345248 = W3  
0.0001000000000000 >= W3

0.0000002600000000 <= W4  
0.0000593257272715 = W4  
0.0001000000000000 >= W4

0.0000002600000000 <=W6=W11  
0.0000609471803244 = W6=W11

0.0001000000000000 >=W6=W11

0.0000002600000000 <=W7=W8

0.0000686805618401 = W7=W8

0.0001000000000000 >=W7=W8

0.0000002600000000 <= W10

0.0000681999417402 = W10

0.0001000000000000 >= W10

0.000000000100000 <=C

0.000000000406752 = C

0.000000000100000 >=C

O arquivo seev.cir foi gravado com os dados para entrada no programa  
SMASH v5

Grato por usar este programa.

Ricardo Silva

SILVA, Ricardo Almeida Araujo da, Otimização de Projetos de Circuitos Integrados Analógicos Usando Métodos de Pontos Interiores em Funções Não Convexas e Não Lineares no Ambiente MATLAB. 2011 1 f. Dissertação (Mestrado em Engenharia Elétrica), Escola Politécnica, Universidade Federal da Bahia, Salvador.

Autorizo a reprodução total ou parcial deste trabalho para fins de comutação bibliográfica

Salvador, 8 de julho de 2011

Ricardo Almeida Araujo da Silva