Supplemental Manual for PISCES-2HB

Circuit Boundary Conditions and Harmonic Balance Module

Francis Rotella, Boris Troyanovsky, Zhiping Yu, and Robert Dutton

Integrated Circuits Laboratory
Stanford University
Stanford, California 94305

Table of Contents

Acknowledgments

The authors would like to acknowledge the following industrial partners who have provided input and advice in the development of various parts of this code (PISCES-2HB):

- Hewlett Packard Co., Santa Rosa, CA
  harmonic balance solver development.
- Motorola, Tempe, AZ
  harmonic balance simulation of RF devices.

The authors would like to acknowledge the continued support of the Semiconductor Research Corporation (SRC) through contract SRC 97-SJ-116.

Foreword

The purpose of this document is to provide a description of advancements made to PISCES to incorporate circuit components into the device simulation and to provide the large signal analysis capabilities for the mixed device and circuit simulation. This simulator is targeted to use by RF device design engineers for analysis and optimization of device structures to achieve peak RF performance.

This manual is divided into four chapters. The first two chapters summarize the implementation of the harmonic balance (HB) solver and the circuit boundary conditions. The details of these additions to PISCES can be found in the dissertations by Troyanovsky and Rotella (references for both can be found in the bibliography [1]-[2]). The third chapter lists the new PISCES cards and parameters in order to access the new capabilities of PISCES. The last chapter provides examples from simple ones to demonstrate the capabilities of the simulator (an expansion of these examples can be found in the dissertations by Troyanovsky and Rotella) to a more in-depth industrial RF design example developed in conjunction with Motorola (more details are available in the dissertation by Rotella).

It is the authors' hope that these additions and advancements in PISCES provide tools and capabilities in the realm of RF so that optimized designs and structures can be generated.

CHAPTER 1
Harmonic Balance Solver

1.1 Introduction

The harmonic balance solver is an addition to PISCES provided by the EEsof Division of Hewlett Packard Co. The solver is designed to interface with Stanford's PISCES-2HB to provide the numerical capabilities to solve the harmonic balance (HB) system of equations generated from PISCES. This section discusses the basic of the solver and the integration with PISCES. For a full description of algorithms and techniques for solving the set of HB device equations, please refer to the dissertation, *Frequency Domain Algorithms for Simulating Large Signal Distortion in Semiconductor Devices*, by Boris Troyanovsky [1].

1.2 Harmonic Balance Basics

Stanford and HP jointly developed a PISCES extension that solves the semiconductor equations in the frequency domain (harmonic bBalance simulation) thus enabling large signal sinusoidal simulations. In addition, a mixed circuit/device simulation capability allows the inclusion of circuit aspects from parasitics to matching networks has been developed to complement the harmonic balance solver. Both are integral parts of the device simulation tool for high frequency, large signal simulation of semiconductor devices [3].

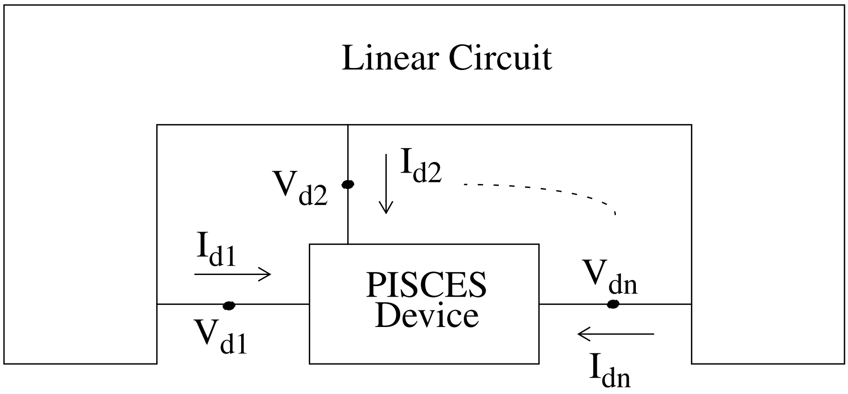


Figure 1.1 PISCES device surrounded by an arbitrary linear circuit

The harmonic balance approach takes the semiconductor equations and assumes periodic solutions given that the input signal is periodic or quasi-periodic. Since the device is nonlinear, integer-multiple combination of harmonics to the input frequency(ies) are created. In simulation, the frequency list is truncated to some set of N frequencies such that the excluded values are insignificant.

The independent semiconductor variables, ψ, n, p, at each grid point are represented as shown in Equation 1.1 where k refers to the solution at a specific grid point in the spatial discretization and i represents the i-th frequency.

$$\psi_k(t) = \psi_{k,0} + \sum_{i=1}^{N} [\psi_{k,i}\cos(\omega_i t) + \psi_{k,i+N}\sin(\omega_i t)] \tag{1.1}$$

The goal of the analysis is to find the quasi Fourier coefficients ψ_{ki}, n_{ki}, and p_{ki} such that the semiconductor partial differential equations are satisfied. Once those coefficients (i.e. harmonics) for the basic variables are obtained, the conduction current and displacement current can be computed at the device terminals.

To include the circuit components in the device simulation they are reduced to a set of boundary condition equations. Figure 1.1 shows a PISCES device surrounded by an arbitrary linear network. By assuming unknown voltages and currents at the electrodes of the device, the relationship among those

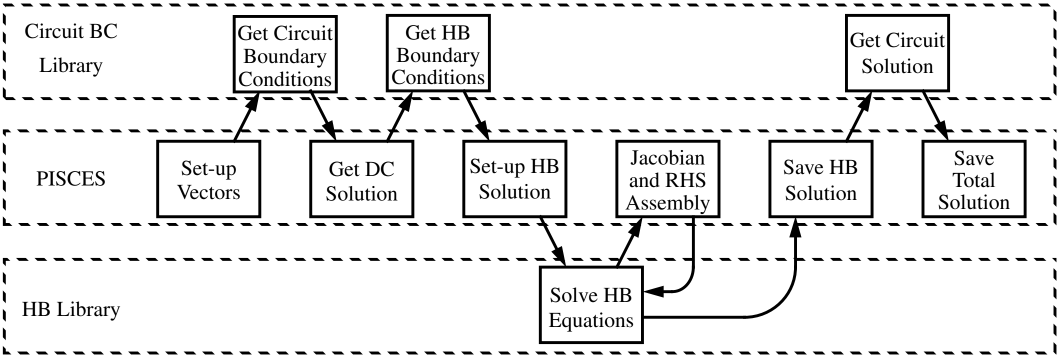


Figure 1.2 Integration of PISCES with a harmonic balance library and a circuit boundary condition library.

voltages and currents are computed in Equation 1.2. For large signal AC analysis, the boundary conditions become complex and are calculated at each frequency. The $V_{applied}$ vector contains the source vector at DC and the fundamental frequencies. It is zero for the generated harmonics since the surrounding circuity is linear and hence generates no new harmonics. For a more in-depth discussion of the circuit boundary conditions for harmonic balance, please refer to Chapter 2.

$$G(V_{applied} - V_{device}) - I(\psi, n, p) = 0 \tag{1.2}$$

Circuit boundary conditions and harmonic balance simulation are integrated with PISCES as shown in Figure 1.2. PISCES calls the circuit boundary condition routines to generate the equations from a SPICE like netlist. It then calls the harmonic balance solver passing the boundary conditions and PISCES state variables. The harmonic balance solver uses the PISCES assembly routines to generate the Jacobian and RHS at each frequency for each iteration. Once the convergence criteria are met, the solution is passed back to PISCES which calls a circuit solver to get the complete solution.

CHAPTER 2
Boundary Conditions for Linear Circuit Elements

2.1 Description

Some simple, yet important circuits consist of only one nonlinear device and the remainings of the circuit are linear circuit components surrounding the nonlinear device. This chapter discusses a method of including those components with a device simulation by reducing the circuit to a set of boundary conditions based upon standard circuit techniques [4]-[5]. The basics are explained with DC analysis and are then expanded to include AC, transient, and harmonic balance analyses.

2.2 Boundary Condition Equations (DC Analysis)

Most device simulators allow external resistances on a contact such that

$$G(V_{app} - V_{con}) - I(\Psi, n, p) = 0 \tag{2.1}$$

where G is the conductance, V_{app} is the applied voltage, V_{con} is the contact voltage, and I is the current which is a function of the semiconductor variables for that specific electrode.

For a linear circuit around a device, the previous equation can be expressed in a matrix format. The voltages and currents become vectors and G becomes a matrix. Both G and V_{app} are constants. Hence,

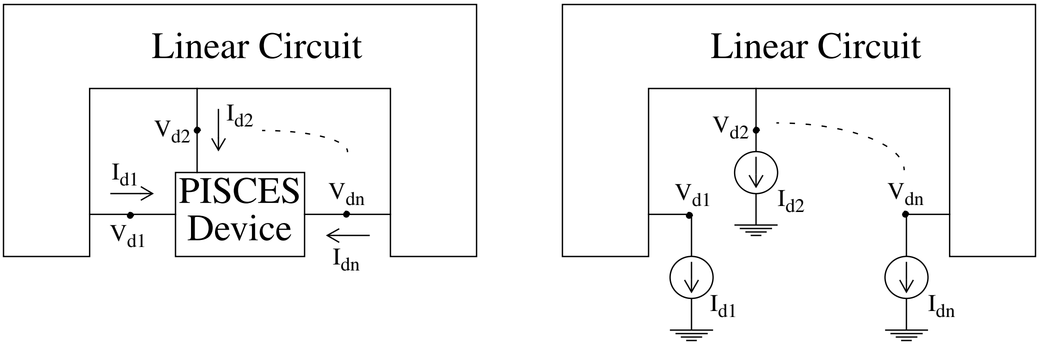


Figure 2.1 (a) Linear circuit surrounding a numerical device. (b) For the purpose of finding the boundary conditions, the numerical device is assumed to consist of ideal current sources of unknown values.

the boundary condition on a node is dependent on the voltages on all nodes to which it is connected via superposition.

In order to determine this matrix equation, consider Figure 2.1 where we have a linear circuit surrounding a numerical device. Substituting ideal current sources for the numerical device connections, modified nodal analysis (MNA) may be applied to the system with the assumption that the device currents are unknown:

$$G_c V_c = I_c + I_d' \tag{2.2}$$

where G_c is the circuit MNA conductance matrix, V_c is the unknown circuit node voltages, I_c is the known circuit currents, and I_d' is the unknown device currents. I_d' is zero for nodes to which the device is not connected. Hence, for a three contact numerical device, this vector only has three entries.

Solving this equation yields Equation 2.3 and re-arranging yields Equation 2.4. Within this system lays the boundary conditions for the numerical device as given in Equation 2.1 simply by re-arranging the equation.

$$V_c - G_c^{-1} I_d' = G_c^{-1} I_c \tag{2.3}$$

$$V_d + R_d I_d = E_d \tag{2.4}$$

E_d can be found by LU decomposing G_c and then using backward substitution. Essentially, the linear circuit is solved for $\left[I_d'\right] = \left[0\right]$. E_d is then just a subset of that solution for the circuit nodes connected to the numerical device.

Since I_d' is symbolic, this derivation requires the computation of G_c^{-1}, which is computationally inefficient. A better approach is to calculate R_d by perturbing the circuit solution for $\left[I_d'\right] = \left[0\right]$ at the circuit nodes to which the device is connect. This process is represented in Equation 2.5:

$$r_{d_{ij}} = \frac{dv_{d_{ij}}}{di_{d_{ij}}} = \frac{\Delta v_{d_{ij}}}{\Delta i_{d_{ij}}} \tag{2.5}$$

Hence, no inverse is needed and a simple LU decomposition of G_c with a few backward substitutions yields Equation 2.4. To clarify notations, Equation 2.1 and Equation 2.4 are compared to find that $V_d = V_{con}$, $E_d = V_{app}$, $G = R_d^{-1}$, and $I_d = I(\Psi, n, p)$.

The boundary condition equations are loaded into the PISCES matrices to force the solution to meet those conditions. When those conditions are met, I_d' is determined so that the entire circuit matrix can be solved for the values at each circuit node.

2.3 AC Analysis

In AC analysis of the circuit, all non-linear components are linearized and the resulting system is solved. A similar approach is followed with circuit boundary conditions. First a DC solution is computed for the nonlinear device. Upon finding a solution, the PISCES device is linearized using small signal analysis at the frequency of interest [6]. That linearized model replaces the numerical device and a simple circuit solve produces the small signal response.

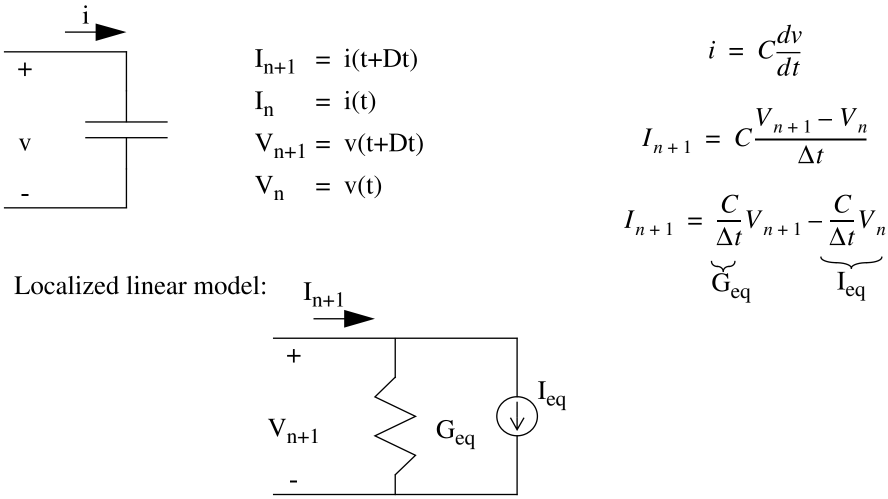


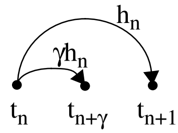
Figure 2.2 Time discretization of a capacitor using Backward Euler.

2.4 Transient Analysis

Transient analysis presents an additional dimension because of the time discretization, but since derivatives are linear the boundary conditions are still obtainable. The two methods of time discretization used in device simulations tend to be Backward Euler (BE) and the TR/BDF2 method of Bank, etc. [7]. In the surrounding circuitry, the only two time dependent elements are capacitors and inductors. For each, the derivative is locally linearized and the resulting system is solved via MNA.

Referring to Figure 2.2, the voltage across a capacitor is known from the previous solution at step t_n. Given the IV relationship for a capacitor, a Backward Euler approximation for the derivative may be employed. Upon simplification, one notes that the resultant equation is equivalent to a current source and resistor in parallel. Hence, for each time step the capacitors can be reduced to two linear elements. As a result, the boundary condition equations can be assembled in the same manner described previously.

DIFF. EQ: $$\frac{d}{dt}(q(z(t))) + f(t, z(t)) = 0$$



TR STEP: $$2q_{n+\gamma} + \gamma h_n f_{n+\gamma} = 2q_n - \gamma h_n f_n$$

BDF2 STEP: $$(2-\gamma)q_{n+1} + (1-\gamma)h_n f_{n+1} = \gamma^{-1}q_{n+\gamma} - \gamma^{-1}(1-\gamma)^2 h_n q_n$$

Figure 2.3 The TR/BDF2 method of Bank etc. for solving a time dependant differential equation.

Note that G_{eq} remains unchanged if the change in time step is constant. As a result, the MNA matrix G_c is unchanged as well as the conductance matrix for the boundary condition equations. Hence, the only term that needs to be recalculated is E_d. For inductors, a similar result is obtained.

In order to have time step control in device simulation, Bank etc. developed a two step method for using a trapezoidal step (TR) followed by a second order backward difference step (BDF2) [7]. Given a differential equation as shown in Figure 2.3 and a time step h_n, a TR step is taken for γh_n and is followed by a BDF2 step of $(1-\gamma)h_n$. If $\gamma = 2 - \sqrt{2} \approx 0.586$ only one Jacobian is necessary for both time steps.

In applying the Bank method to the differential equations for inductors and capacitors yields a set of locally linearized model similar to the backward Euler method. Note that G_{eq} is the same for both the TR step and BDF2 step given the requirement that $\gamma = 2 - \sqrt{2} \approx 0.586$. Likewise, it is the same if h_n is the same. As a result, in both these situations, only E_d needs to be recomputed with each time step.

2.5 HB Analysis

Harmonic balance simulation presents another special case for circuit boundary conditions. As explained in Chapter 1, harmonic balance is used to solve the semiconductor differential equations by assuming a sinusoidal solution at integer multiples of the fundamental frequency(ies). Hence, a boundary condition equation is required for each frequency. Since the circuit is linear, the equations are straightforward because no additional harmonics are generated in the circuit and the set of complex boundary condition equations fall into the following three categories:

For the DC bias, the equation is determined by the methodology described previously.

$$G_{d0}(E_{d0} - V_{d0}) - I_{d0}(\Psi_0, n_0, p_0) = 0 \tag{2.6}$$

For frequencies at which there is a source (i.e. the fundamental(s)) the equation is given in a similar format except that it is complex.

$$Y_{d1}(E_{d1} - V_{d1}) - I_{d1}(\Psi_1, n_1, p_1) = 0 \tag{2.7}$$

Finally, for the harmonics generated by the nonlinear PISCES device, the equation takes on the following form. Note that there is no generation term E_{dn}

$$-Y_{dn}V_{dn} - I_{dn}(\Psi_n, n_n, p_n) = 0 \tag{2.8}$$

The harmonic balance module computes the magnitude and phase of the current at each electrode of the device for each frequency. That solution is then used in the circuit equations to compute the final circuit solution.

2.6 Compiling PISCES with Circuit Library

The current release of PISCES contains the source code for the circuit boundary conditions and should be configured to be compiled with PISCES. This section briefly informs the user about the details of the compilation.

The source code for the library is located in the `cktlib` in the PISCES `src` directory. Typing `make` in this directory will create the library `libckt.a` in the directory `lib`. To compile the library with PISCES, go to the `pisces` directory and edit the `Makefile` there. In the `LOCLIBS` definition add the library `-lckt`.

In addition, the circuit library requires the sparse matrix package from the University of California at Berkeley. This public domain library is included in the directory `sparse` already configured for use with PISCES. Compiling the code will generate the `libsparse.a` library in the directory `lib`. Likewise, include this library on the `LOCLIBS` definition with a `-lsparse`. Note that the harmonic balance library could also contain a version of sparse. As a result, if sparse routines are included in the HB library on the `LOCLIBS` definition, then it is not necessary to include it as a separate item.

CHAPTER 3
Users Manual

3.1 Introduction

New parameters and new cards were added to the PISCES in order to access the new capabilities. In addition, for circuit boundary conditions, a SPICE like input deck is required to describe the circuit configuration. This chapter describes the new PISCES cards and parameters as well as the circuit description input netlist.

3.2 PISCES Card Additions

The improvements from PISCES-2H to PISCES-2HB involves additions for harmonic balance simulation, circuit boundary conditions, new models, and improved numerics. Each change results in new parameters in order to invoke the capability. In addition, harmonic balance simulation necessitates the additional card. `hbmeth` is used to define the parameters for the numerics involved in harmonic balance simulation and is similar to the `method` card. In addition, a separate file containing a SPICE-like netlist is required to describe the circuit surrounding the PISCES device. The format for this file is described after the new PISCES parameters and cards are catalogued.

COMMAND CONTACT

The new parameter on this card allows for the specification of the input file containing the circuit boundary conditions.

SYNTAX

Contact cktfile = <filename>

NEW PARAMETERS

cktfile

This parameters specifies the input file that contains the netlist for the linear circuit surrounding the device.

EXAMPLE

```
contact cktfile=myckt.ckt
```

COMMAND HBMETH

This new card specifies numerical limits and methods for the harmonic balance module. It is similar in nature to the **method** card.

SYNTAX

hbmeth [maxiter = <integer>] [useredpc] [krtol = <real>]
+ [kriter = <integer>] [gmrestart = <integer>]
+ [pkthresh = <real>] [gmrthresh = <real>]
+ [krloosetol = <real> krlooseiter = <integer>]
+ [krconvratio = <real>]

PARAMETERS

maxiter

This parameter specifies the maximum number of Newton iterations before the HB simulator aborts and steps the sources down to try again. A reasonable value is 30 although a good number is not known in practice (Default: 30).

useredpc

This logical variable is used to specify the ‘reduced’ pre-conditioner. This parameter is needed only for two-tone problems with very tightly or very widely spaced tones and reduces memory dramatically (Default: false).

krtol

This variable specifies the tolerance for the iterative linear (Krylov) solves during the Newton process. Reasonable values are 1 x 10^{-3} and 1 x 10^{-4} (Default: 1 x 10^{-4}).

kriter

This parameter specifies the maximum number of iterations during the iterative linear solve before it gives up and just takes the Newton step anyway. A value of 30-50 is reasonable (Default: 40).

gmrestart

This integer value is the number of iterations that GMRES will take before "restarting." In theory, the higher this number, the faster and more robust the linear solve will be. However, an additional vector has to be stored for each GMRES iteration, so its value cannot be too large. 10 tends to work well (Default: 10).

pkthresh

This parameter is used to reduce Jacobian memory storage for large problems. Spectral components less than (**pkthresh** x DC_value) are dropped (Default: 0.0).

gmrthresh

This parameters is used to reduce GMRES memory storage for large problems. GMRES vector entries less than **gmrthresh** are dropped (Default: 0.0).

krloosetol

This parameter is used jointly with **krlooseiter**. If **krlooseiter** GMRES iterations are completed and the GMRES error is below **krloosetol**, then the GMRES iterations are considered adequate for proceeding with a Newton step (Default: 0.1).

krlooseiter

This parameter is used jointly with **krloosetol**. If **krlooseiter** GMRES iterations are completed and the GMRES error is below **krloosetol**, then the GMRES iterations are considered adequate for proceeding with a Newton step (Default: 30).

krconvratio

When the ratio between two successive GMRES reaches the value of this parameter, GMRES assumes progress toward a solution is stalled and therefore, aborts (Default: 1.0).

EXAMPLE

```
hbmeth krtol=5e-4 kiter=35 ^useredpc
```

COMMAND LOG

The new parameters on this card allows for the storage of the circuit solution independent of the device solution.

SYNTAX

log cktfile=<filename> [column.out | spice.out | key.out]

NEW PARAMETERS

cktfile

This parameter specifies the output file name for the circuit solution. It contains the voltage at each circuit node, the current through all voltage sources and inductors, and the current flowing into each node of the PISCES device including any scaling factors (Default: No data saved).

column.out

This flag forces the output data into columns with a heading describing the data in columns. It can be post processed by many generic tools and spreadsheets (Default: True).

spice.out

This flag forces the data to be saved in a Berkeley SPICE-like raw file format. For those with access to Berkeley's plotting tools, this format provides an easily accessible tool to analyze the circuit solution. **spice.out** is only available for **opckt**, **acckt**, **dcckt**, or **trckt** (Default: False).

key.out

This flag forces the output data into two files. One file has an extension of **.dat** add to the **cktfile** name and the second has an extension of **.key**. The **.dat** file contains the simulation results in columns. The **.key** contains a mapping of the solution variables to the column number in the **.dat**. This output format allows the user to plot the data in many generic tools by excluding the heading information in the data file which can cause problems with some plotting tools (Default: False).

EXAMPLE

log cktfile=mydata.dat column.out

log cktfile=mydata.raw spice.out

log cktfile=mydata key.out

COMMAND SOLVE

Many parameters are added to the **solve** card in order to take advantage of the new capabilities in PISCES. A new set of parameters are provided to invoke the circuit boundary condition and the harmonic balance solution. In addition, solution input/output parameters are provided for the harmonic balance results.

SYNTAX

solve [opckt | acckt | dcckt | trckt | acdcckt | hbckt]
+ [hboutfile=<filename> [savestep=<integer>]]
+ [hbinfile=<filename>]

PARAMETERS

opckt

Solves for the operating point on the circuit (Default: False).

acckt

Solves the small signal AC circuit. This parameter requires an **.ac** card in the circuit file (Default: False).

dcckt

Solves the DC circuit for the values of the swept source(s) specified on the **.dc** card in the circuit file (Default: False).

trckt

Solves for the transient response of the circuit. This card requires a **.tran** card in the circuit file. In addition the **tranckt** may be used in lieu of **trckt** (Default: False).

acdcckt

Solve the small signal AC circuit while sweeping DC source(s). This card requires **.dc** and **.ac** cards in the circuit file. In addition **dcacckt** may be used in lieu of **acdcckt** (Default: False).

hbckt

Solves the large signal AC circuit using the harmonic balance simulator. This card requires the HB simulator module and the **.hb** analysis card in the circuit file. A sweeping of the DC and/or AC source can be specified with the **.hbac** and/or the **.hbdc** card in the circuit file (Default: False).

hboutfile

The base part for the name of the file to store the solution at a given simulation point. To this name an extension of "**.sol**" is added for the data for Ψ, n, and p. An extension of "**.trm**" is added to the name for the data file containing the terminal characteristics. A "**.nds**" extension is added for the data file containing the circuit node solutions.

savestep

During a sweep of a source in a harmonic balance simulation, **savestep** is used to specify how often the harmonic balance solution is saved given a file base part specified by **hboutfile**. At each integer multiple of the given value, the last letter of the output file name is incremented and the solution is saved to that file. A value of 0 means only the last completed solution is saved (Default: 0).

hbinfile

Use the given file name as the starting point for the next harmonic balance simulation. This file contains the "**.sol**" extension and it must be specified when the file name is given.

NEW EXAMPLE

```
solve acckt

solve hbckt hboutfile=hbsolnA hbsavestep=15
```

3.3 The Circuit File

A separate circuit file is needed to specify the linear circuit surrounding the PISCES device. The circuit file contains a SPICE-like netlist to describe the connections of the external circuitry [4]. The standard SPICE linear elements are provided along with a special element for specifying the electrode connections for the linear devices. The standard analysis capabilities are provided with the **.op**, **.dc**, **.ac**, and **.tran** cards. In addition there are specialty dot ("**.**") cards for the harmonic balance analysis.

The **contact** card in PISCES contains a parameter which is used to specify the file name for the circuit file. Even if a circuit file is specified for boundary conditions, the boundary conditions are not invoked unless one of the circuit solves is specified on the **solve** card. Therefore, the recommended sequence of cards in PISCES is as follows:

```
solve init outfile=soln.ini
solve v1=1.0 vstep=1.0 nstep=4 num=1
solve v2=0.25 vstep=0.25 nstep=5 num=2
solve v1=5.0 v2=1.5 outfile=soln.dcini
solve opckt outfile=soln.op
solve trckt
```

In this sequence, an initial solution is computed and saved in the file `soln.ini`. The next card ramps up the voltage on electrode number one, which could be a drain or collector. The third `solve` card ramps up the voltage on electrode two, which could be a gate or base. The fourth card solves for the approximate operating point and stores that solution in file `soln.dcini`. The fifth `solve` card accesses the circuit boundary conditions and solves for the operating point solution with the inclusion of circuit boundary conditions. The final `solve` card specifies a transient simulation with circuit boundary conditions. It could just as easily specify an ac, dc, or harmonic balance circuit analysis. When the simulation is restarted or fails, one can now use the `load` card to restart from anywhere solution has been saved for.

The next few pages provides description of the elements and analyses cards that can be specified in the circuit file. At the end of the chapter some limitation on node numbers and information on value specifications are outlined.

COMMAND COMMENT LINES

Any line with an * (asterisk) in the first column is considered a comment and is ignored during parsing.

```
* This is a comment
```

COMMAND NUMERICAL DEVICE

Nxxxxxxxx N1 . . . Ni

The PISCES device is specified by the numerical device element card. **xxxxxxxx** uniquely identifies the device and **N1** through **Ni** are the numerical nodes to which the device is connected in the circuit. The number of node connections must equal the number of electrodes defined in the PISCES deck, otherwise the simulation will abort.

Nldmos 14 3 0

COMMAND STANDARD ELEMENTS

Rxxxxxxxx N1 N2 val
Lxxxxxxxx N1 N2 val
Cxxxxxxxx N1 N2 val

The **R**, **L**, and **C** identify the element as either a resistor, an inductor, or a capacitor, respectively. **N1** and **N2** are the numeric node numbers to which the element is connected. The **val** variable represents the value of that element. The standard MKS units can be used to specify values and are described later in this document.

```
Rfeeback 23 81 1e3
Lpackage 87 63 93n
Cintercon 9 38 2.33p
```

COMMAND TRANSMISSION LINE

Txxxxxxxx N1 N2 GND R L C G Len Nsections

This element specifies a two port (common ground) distributed T-Network transmission line as shown in the figure. **N1** and **N2** are the numeric node numbers at each end of the line and **GND** is the common ground contact. The distributed nature of the line is created by T-sections where **R**, **L**, **C**, and **G** specify the resistance per unit length, the inductance per unit length, the capacitance per unit length, and the admittance per unit length. **Len** is the total length of the line in unit lengths and **Nsections** is the number of T-sections used to simulate the line. Hence, the inductance of each T-section is given by **l** = **L*****Len**/(**2*****Nsections**), the resistance is given by **r** = **R*****Len**/(2***Nsections**), the capacitance is given by **c** = **C*****Len**/**Nsections**, and the admittance is given by **g** = **G*****Len**/**Nsections**. Note that the circuit portion of any simulation is very small compared to the device simulation. Hence, a relatively large number of sections not only improves the transmission line model, but also does not realistically affect the device simulation.

Tmatch 8 10 0 0 28n 100p 0 1 200
Tmatch 7 9 0 0 14n 50p 0 2 200
Tinput 2 3 0 1u 82n 54p 3n 5 100

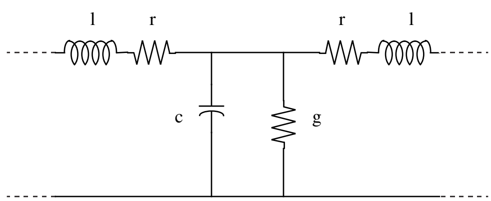


Figure 5.1 One T-section of a transmission line defined with N sections.

COMMAND DEPENDENT SOURCES

Gxxxxxxxx N+ N- n+ n- val
Exxxxxxxx N+ N- n+ n- val

G and **E** specify a linear voltage controlled current source and a linear voltage controlled voltage source. **N+** and **N-** specify the positive and negative nodes of the source. Current flows from the positive node to the negative node. **n+** and **n-** specify the voltage nodes which control the sources. **val** is the transconductance (in mhos) or the voltage gain respectively.

Fxxxxxxxx N+ N- Vname val
Hxxxxxxxx N+ N- Vname val

F and **H** specify a linear current controlled current source and a linear current controlled voltage source. **N+** and **N-** specify the positive and negative nodes of the device. Current flows from the positive node to the negative node. **Vname** specifies the voltage source whose current controls the devices. **val** is the current gain or transresistance (in ohms) respectively.

```
Gcond 0 12 20 0 1m
Egain 10 0 20 0 50
Fgain 0 40 VinCur 3
Ftran 23 0 VoutCur 1k
```

COMMAND INDEPENDENT SOURCES

Vxxxxxxxx N+ N- <source description>
Ixxxxxxxx N+ N- <source description>

V and **I** specify an independent voltage source and current source, respectively. **N+** and **N-** are the numerical nodes to which the source is connected. Current flows from the positive node to the negative node through the source.

The source descriptions are as follows. A voltage source is used for reference, but all descriptors can be used with a current source.

DC Sources

Vname 1 0 [dc] [value]

The **dc** specifies the source as a DC source and the value is the given source value. Neither **dc** nor **value** needs to be specified. If only **value** is specified then the source is assumed to be a DC source of that value. If neither **dc** nor **value** or if only **dc** is specified then the source is assumed to be of DC value zero.

Vgate 5 0 dc 2.3

Vcc 1 0 5.0

AC Sources

Vname 1 0 [dc] [DCvalue] ac [ACvalue]

The **ac** specifies this source as being AC and is required on all AC sources. **ACvalue** specifies the amplitude of the AC small signal generated by the source. If no value is given, **ACvalue** is assumed to be 1.0. An optional **dc** specification may be placed on the source to indicate that it also supplies a DC bias.

Vperturb 5 0 dc 2.3 ac 0.1

Vinput 9 0 ac 0.6

TRANSIENT SOURCES

Vname 1 0 <transient source type>

Transient sources can take on a number of different function as described by the following functions.

Piece-wise Linear (pwl)

Vname 1 0 pwl t0 v0 t1 v1 t2 v2 t3 v3 . . .

Piece-wise linear sources are described by a discontinuous function where each (time, voltage) point is connected by a linear change in the value of the source. The points in the function are given as:

Time	Value
t0	v0
t1	v1
t2	v2
...	...

if t0 != 0 then the voltage at t=0 is set equal to the voltage at t = t0.

Vinput 10 0 pwl 0 0.0 1n 0.0 2n 0.5 10n 2.0 20n 2.0 21n 0.0

Periodic Square Pulse (pulse)

Vname 1 0 pulse v1 v2 td tr tf pw per

A **pulse** is simply a piece-wise linear function specified in an easier manner to define a periodic function. The variables are defined as:

Variable	Description
v1	initial value
v2	pulsed value
td	time delay
tr	rise time
tf	fall time

Variable	Description
pw	pulse width
per	period

and the function takes on the following values as a piece-wise linear function:

Time	Value
0.0	v1
td	v1
td+tr	v2
td+tr+pw	v2
td+tr+pw+tf	v1
per+td	v1
per+td+tr	v2
...	...

Vname 1 0 pulse -0.5 0.5 10n 1n 1n 50n 100n

Sinusoidal Function (sin)

Vname 1 0 sin Vo Va freq td theta

This source takes on the following sinusoidal functional values within the time ranges specified. Note that this does not define a source for harmonic balance, but rather defines a source for transient analysis.

$0 < t < \mathrm{td}$	Vo
$\mathrm{td} < t$	$\mathrm{Vo} + \mathrm{Va}\, e^{(-(t-\mathrm{td})\mathrm{theta})} \sin(2\pi\ \mathrm{freq}\ (t + \mathrm{td}))$

where:

Variable	Description
Vo	DC offset value
Va	sinusoid amplitude
freq	frequency in Hz
td	delay time
theta	damping factor

Vname 1 0 sin 3.3 0.5 50Meg 0.0 0.0

Exponential Pulse (exp)

Vname 1 0 exp v1 v2 td1 tau1 td2 tau2

The exponential source takes on the following exponential functional values within the time ranges specified.

$$0 < t < \mathrm{td1} \qquad \mathrm{V1}$$

$$\mathrm{td1} < t < \mathrm{td2} \qquad \mathrm{V1} + (\mathrm{V2} - \mathrm{V1})\left(1 - e^{\left(-\frac{-(t - \mathrm{td1})}{\mathrm{tau1}}\right)}\right)$$

$$\mathrm{td2} < t \qquad \mathrm{V1} + (\mathrm{V2} - \mathrm{V1})\left(1 - e^{\left(-\frac{(t - \mathrm{td1})}{\mathrm{tau1}}\right)}\right) + (\mathrm{V1} - \mathrm{V2})\left(1 - e^{\left(-\frac{(t - \mathrm{td2})}{\mathrm{tau2}}\right)}\right)$$

where:

Variable	Description
v1	initial value
v2	pulsed value
td1	rise delay time
tau1	rise time constant

Variable	Description
td2	fall delay time
tau2	fall time constant

Vname 1 0 exp -1.5 0.75 3n 25n 50n 10

Single Frequency FM (sffm)

Vname 1 0 sffm vo va fc mdi fs

The sffm source takes on the following single frequency FM functional value over all time:

$$\mathrm{Vo} + \mathrm{Va} \sin(2\pi\ \mathrm{fc}\ t + \mathrm{mdi} \sin(2\pi\ \mathrm{fs}\ t))$$

where:

Variable	Description
Vo	DC offset voltage
Va	amplitude
fc	carrier frequency
mdi	modulation index
fs	signal frequency

Vname 1 0 sffm 0.0 1m 108Meg 5 40k

Harmonic Balance Source

Vname 1 0 hb Vo Vmag freq theta

Harmonic balance source is used to define the large signal AC source applied to the circuit during harmonic balance analysis. Note that a sinusoidal transient source is ignored during harmonic balance analysis and likewise, a harmonic balance source is ignored during transient analysis. All AC sources in the HB simulation must be specified with this source type. The parameters are defined as follows:

Variable	Description
Vo	DC offset voltage
Vmag	amplitude of sinusoid
freq	frequency of sinusoid
theta	phase of sinusoid (deg.)

Hence the source takes on the following values for the given frequencies:

Frequency	Value
DC	Vo
freq	$\mathrm{Vm}\cos(\mathrm{theta}) + j\mathrm{Vm}\ \sin(\mathrm{theta})$
$\mathrm{n} \times \mathrm{freq}$	0.0

Vname 1 0 hb 0.0 0.5 1.2G -90

COMMAND .OPTIONS CARD

The **.options** card is used to set numerical parameters related to the circuit simulation.

SYNTAX

.options [gmin=value] [rmin=value] [area=value] [ltertol=value] [ltevtol=value] [lteitol=value] [ltelim=value]

PARAMETER

gmin

The **gmin** parameter specifies the smallest value for a conductance used in the circuit simulation. A **gmin** conductor is placed from every node to ground. This additional large resistance adds a leakage current, but guarantees that the matrix is not singular. Note that too small a value of **gmin** may cause ill-conditioning. For most applications, **gmin** could be set to zero and hence, have no affect. However, if there are very few or no passive components in the circuit, **gmin** has to be larger than zero. (Default: 1.0E-12 mhos).

rmin

The **rmin** parameter is used to specify a small resistance for places in the circuit where a small short-circuit is required. This resistance is typically used for an inductor connecting a voltage source solely to an electrode of the PISCES device. Inductors are replaced with a zero volts source during DC analysis and hence, a zero resistance in this situation can causes the matrices to go singular. Like **gmin**, the effect of **rmin** is minimal. Only a small voltage is typically lost across the resistor and it guarantees that the matrix is not singular. The **rmin** parameter can be set to zero for most cases, but should a matrix inversion error occur, this parameter should be set to some small inconsequential value (Default: 1.0E-12 ohms).

area

PISCES solves a device in only two dimensions. The area parameter is a scaling factor for all currents in order to provide a quasi-3D solution. For most applications, this parameter refers to the width of the device although symmetry (like in BJT's) could mean **area** is equal to twice the device length or more.

ltertol, ltevtol, lteitol, ltelim

These four parameters are used in the calculations of the local truncation error during transient analysis. For most applications, the default values for these parameters are sufficient, but a brief description is provided for a user who may want to adjust the time step controls of the circuit portion of the simulation. For a more detailed description, please refer to the paper by Bank, *et al*.

ltertol: The relative tolerance for calculating the errors associated with each time step (Default: 0.001).

ltevtol: The absolute error tolerance for voltages (Default: 1mV).

lteitol: The absolute error tolerance for currents (Default: 1pA).

ltelim: The limit on the value for the norm of the error tolerance. If the norm is greater than this value, the time step is reduced and repeated. If the norm is less than this value, the time step is accepted and the next dt is calculated (Default: 1.0).

In order to increase the accuracy in the overall solution, the relative error tolerance (**ltertol**) can be decreased. Likewise, in order to relax the accuracy, **ltertol** can be increased. In order to affect only the voltage or the current, **ltevtol** or **lteitol** should be adjusted in the same manner as **ltertol**. Finally, in order to tighten the time steps, **ltelim** can be reduced or likewise, to relax the time steps, **ltelim** should be increased.

All these parameters only affect the circuit portion of the transient analysis. In most cases, the device simulation tends to limit the time steps, and hence, these parameters should rarely be changed.

EXAMPLES

```
.options gmin=0.0 rmin=0.0 area=100
```

COMMAND ANALYSIS CARDS

A set of analysis cards provides the user a way to specify the limits and conditions on the different types of analysis. Note that PISCES is used to select the desired analysis via the solve card; hence, multiple analysis cards can exist in one file.

Operating Point Simulation

.op

Compute the operating point solution.

DC Sweep Simulation

.dc Sname1 start1 end1 step1 [Sname2 start2 end2 step2]

The **.dc** card allows for the sweeping of DC sources. Up to two sources may be swept simultaneously where the second source is swept inside the first source. The **.dc** requires at least one source and set of sweep parameters to be specified. The **start** parameters refers to the start value, the **end** parameter refers to the end value, and the **step** refers to the stepping value. The source is stepped until its value is greater than or equal to its end value.

```
.dc Vin 0.0 0.5 0.05
```

AC Simulation

.ac [dec | lin] numsteps startf endf

The **.ac** card specifies a small signal sweep in frequency. All ac sources are swept over the specified frequencies as determined by this card. The sweeping can be linear or logarithmic as specified by the **lin** or **dec** parameter. The **lin** parameter means the **numsteps** are taken linearly from **startf** to **endf**. The **dec** parameter means that **numsteps** are taken logarithmically and there are **numsteps** per decade.

.ac lin 10 1k 10k

.ac dec 5 100k 1G

Transient Simulation

.tran tstep tstop [tstart tmax]

The **.tran** card specifies the time range and time steps for a transient analysis. The **tstep** parameter specifies the initial time step. If the backward Euler method is selected on the PISCES method card, this time step is used throughout the entire PISCES/circuit simulation. If the BDF method is selected on the PISCES method card, then **tstep** is the initial time step and time step estimation is used to select all future values. The **tstop** parameter specifies the time at which the simulation stops. The **tstart** parameter specifies the point in time from which the solution is to be saved. If this parameter is not given, the solution is saved from time equal to zero. The **tmax** parameter is the maximum time step to be taken. If it is not given, the maximum is calculated based upon the value of the stop time. This ratio is set at compilation and has a default value of 0.1.

.tran 0.1n 25n

.tran 0.1n 25n 0.0 5n

Harmonic Balance Simulation

.hb f1 order [f2] [f3] [f4] . . .
.hbdc Sname start end step [Sname start end step] . . .
.hbac Sname start end step [Sname start end step] . . .
.hbfr Sname start end step fnum [Sname start end step] . . .
.hbss nstep {a | d | f} Sname start end {fnum} [{a | d | f} Sname . . .]

These analysis cards are used to specify various types of harmonic balance analysis. The **.hb** card is required and describes the Fourier expansion. The **f1** parameter specifies the first fundamental frequency and is required. The subsequent **f#** parameters specify the higher fundamental frequencies if inter-modulation distortion analysis is to be performed. The **order** parameter is the number of harmonics used in the Fourier expansion.

The **.hbdc**, **.hbac**, **.hbfr**, and **.hbss** card specifies any sweeping of sources. The **.hbdc** causes the HB source **Sname** to have its DC bias swept from the **start** value to the **end** value in **step** steps. The sources listed later on the card are swept inside the sources listed first.

The **.hbac** causes the HB source **Sname** to have its large signal AC value swept in magnitude from **start** value to **end** value in **step** steps. The sources listed later on the card are swept inside the sources listed first.

The **.hbfr** card causes the HB source **Sname** to have it frequency value swept from **start** value to **end** value in **step** steps. In addition, the fundament frequency number **fnum** is adjusted to this value as well.

The **.hbss** allows for simultaneous sweeps of multiple sources such that the specified value is adjusted for each and every source. Therefore, only one value is specified for the number of steps, **nstep**. Each source **Sname** has either its AC magnitude, DC value, or frequency value (**a | d | f**) swept from **start** value to **end** value with the same **nstep** steps. If frequency is swept, then the fundamental value is adjusted as given by **fnum**.

Multiple cards may be contained in the same simulation. For such situations, the sources on **.hbss** are swept inside the source on **.hbfr** which are swept inside the sources on the **.hbac** card, which are swept inside the sources on the **.hbdc** card.

.hb 1.2G 8
.hbac Vin 0.25 4.0 0.25

Sweep source Vin from 0.25 to 4.0 by 0.25. Do a HB simulation at 1.2 GHz with 8 harmonics.

.hb 849.5Meg 5 850.5Meg
.hbss 16 Vin1 a 0.25 4.0 Vin2 a 0.25 4.0

Sweep the magnitude of the AC voltage of `Vin1` and `Vin2` from 0.25 to 4.0 volts in 16 steps. Hence, during each step `Vin1` and `Vin2` take on the same value. This sweep description is designed to do inter-modulation distortion analysis by setting the frequency of `Vin1` to 849.5MHz and that of `Vin2` to 850.5MHz. The **.hb** card specifies a 5th order expansion at the two given frequencies.

3.4 NODE NUMBERS

The node numbers in the netlist must be unique positive integers. Negative numbers and alpha-numeric characters will cause an error.

3.5 SCALING FACTORS

The values in the ne list may be specified in decimal notation (e.g., 1.2), exponential notation (4.3e-5, 10e4) or using a scaling factor (1k). The scaling factors are defined as follows:

Unit	**Symbol**	**Scaling**
tera	t	10^{12}
giga	g	10^{9}
mega	meg	10^{6}
kilo	k	10^{3}
centi	c	10^{-2}
milli	m	10^{-3}
micro	u	10^{-6}
nano	n	10^{-9}
pico	p	10^{-12}
femto	f	10^{-15}
atto	a	10^{-18}

CHAPTER 4
Examples

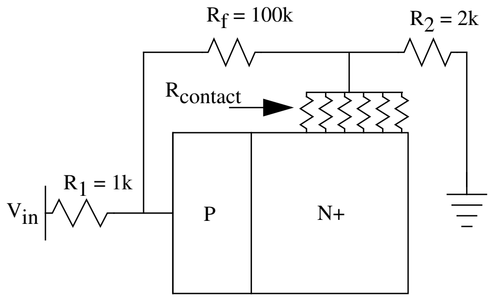
4.1 Description

This chapter provides many examples of using the new features provided in PISCES-2HB. The first section focuses on simulations with external boundary conditions by showing a variety of different analysis methods. The second section gives a number of examples that take advantage of the harmonic balance module to find large signal responses of some common RF circuits including a demodulator, mixer, and power amplifier.

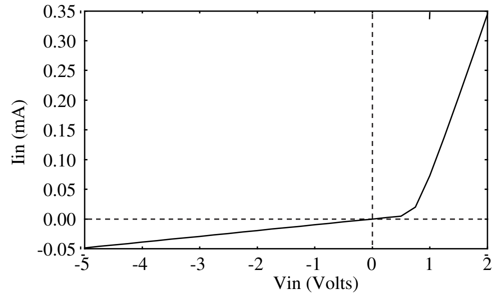
4.2 Circuit Boundary Conditions

4.2.1 Diode with External Circuit Components and Distributed Contact Resistance

PISCES boundary conditions include distributed contact resistances, limited surface recombination, external resistances/capacitances, and linear circuit boundary conditions. This example uses two of these boundary conditions - contact resistance and circuit boundary condition - simultaneously and to plot the I-V characteristics of a diode with external resistances.



(a)



(b)

Figure 4.1 (a) Circuit diagram for diode with extrinsic bulk resistance and distributed contact resistance. (b) The IV response of the structure shows a large leakage current generated by the shunt resistance.

A diode has external resistors on each electrode and a feedback resistor between the two electrodes. In addition, electrode two has a distributed contact resistance as shown in Figure 4.1 Next to the circuit diagram, the IV characteristics are given for the configuration. Note the large leakage current flowing through the shunt resistance.

The input deck for PISCES and circuit description is provided in Figure 4.2a and Figure 4.2b, respectively. The distributed contact resistance is specified on the first **contact** card and the circuit boundary conditions are invoked by specifying a **ckfile** on the second **contact** card. The circuit description is given in the circuit file named `diode.ckt`. The solution for the circuit simulation is stored in `diode.raw` and is in a format that can be read by Berkely's version of SPICE3.

Figure 4.3 contains plots of the current flow lines when the distributed contact resistance is low and high relative to the N- region, respectively. Notice that the current flow changes based upon the magnitude of the distributed resistance.

```
title diode with contact resistance and circuit BC's

$ create mesh
mesh rect nx=70 ny=11

x.mesh n=1 l=0
x.mesh n=70 l=10
y.mesh n=1 l=0
y.mesh n=11 l=0.2

region num=1 ix.lo=1 ix.hi=70 iy.lo=1 iy.hi=11 silicon

$ electrodes
electr num=1 ix.lo=1 ix.hi=1 iy.lo=1 iy.hi=11
electr num=2 ix.lo=50 ix.hi=70 iy.lo=1 iy.hi=1

$ uniform doping for simplicity
doping p.type conc=1e15 uniform
doping n.type conc=2e19 uniform x.left=4

$ place distribute resistance on contact #2
contact num=2 con.res=1e-3'

$ specify circuit file
contact cktfile=diode.ckt

$ set models
model consrh conmob fldmob bgn

$ do initial solution
symb carr=2 newton
method trap
solve init

$ ramp up voltage on diode
solve v1=0 vstep=-1.0 nstep=5 elect=1

$ set circuit logfile in Berkeley spice3f4 format
log cktfile=diode.raw spice.out

$ do dc sweep
solve dcckt
```

(a)

```
* test for multiple boundary conditions

* initial start value for source
vin 1 0 -5

* completely surround device with resistances
r1 1 3 1k
rf 3 4 100k
r2 4 0 2k

* device specification
nd 3 4

* analysis specification
.dc vin -5 2 0.25

* circuit options
.options area=100 rmin=0 gmin=0
```

(b)

Figure 4.2 (a) PISCES input deck for DC sweep of diode surrounded by resistive network with a distributed contact resistance. (b) Netlist describing circuit diagram.

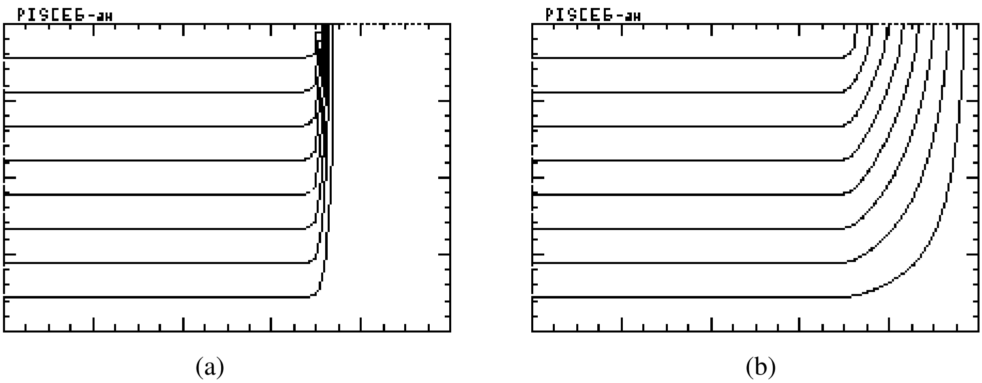


Figure 4.3 Current flow lines for a diode with (a) a low contact resistance and (b) with a large contact resistance.

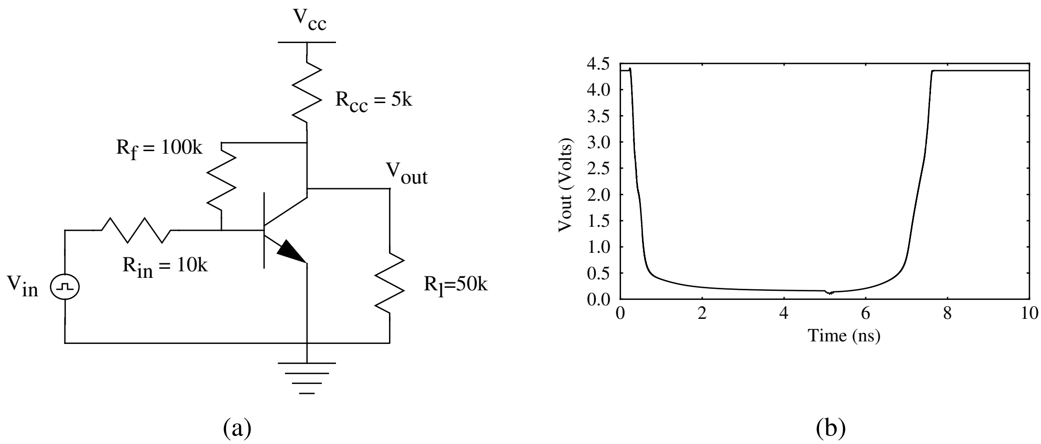


Figure 4.4 Circuit diagram of a simplified BJT inverter and the response of that inverter given a pulse input.

4.2.2 Transient Response of a BJT Inverter

Transient analysis allows for automatic time stepping to minimize the number of time steps that are required. To demonstrate this capability, a simple BJT inverter is switched from its `on` state to its `off` state and back to its `on` state. A circuit diagram and the transient response is provided in Figure 4.4.

```
$ transient analysis of bjt inverter

$ load mesh file
mesh infile=bjt.msh

$ set-up circuit information
contact cktfile=inv.ckt

$ set models for BJT inverter
model srh auger bgn conmob fldmob

$ do symbolic factorization and
$ set numerical methods
symb newton carrier=2
method itlimit=20 biaspart 2nd tauto

$ initial device solution
solve init

$ ramp up voltage on drain
solve v1=0 vstep=1 nstep=5 elect=1

$ solution logfile
log cktfile=inv.raw spice.out

$ do transient analysis of circuit
solve tranckt
```

(a)

```
* Circuit description for a BJT inverter

* apply transient piece-wise linear function at
*   input of the inverter
vin 4 0 pwl 0 0 0.2n 0 0.4n 5 5n 5 5.2n 0 1 0
vcc 1 0 dc 5.0

* surrounding circuitry
rcc 1 2 5k
rbb 4 3 10k
r31 2 3 100k

* load of next stage
rl 2 0 50k

* BJT device
nq1 2 3 0

* set device size
.options area=1

* transient analysis initial step and stop time
.tran 0.1n 10n
```

(b)

Figure 4.5 (a) PISCES input deck for transient simulation of a BJT inverter. (b) Netlist describing circuit diagram.

The PISCES input deck and the circuit file are shown in Figure 4.5. The time stepping algorithm is selected in the PISCES deck on the **method** card with the **2nd** and **tauto** parameter set to true to invoke the TR-BDF method described by Bank. The circuit portion of the code calculates its own time step independent of PISCES. PISCES then decides which time should be taken: its own or that of the circuit where typically the smaller of the two is selected. In addition, PISCES handles break points in the input response by forcing a time point at that specific location.

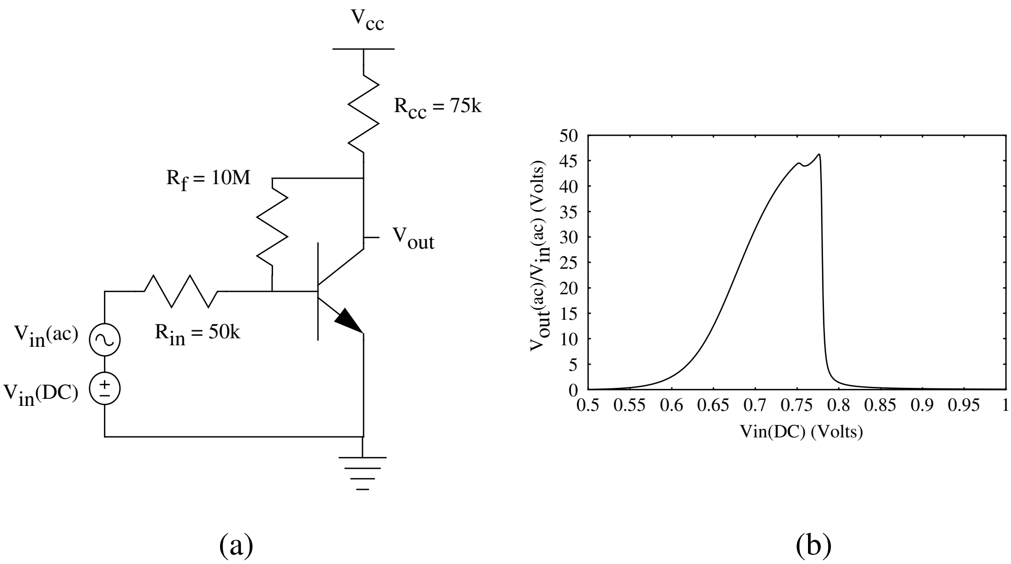


Figure 4.6 (a) Circuit diagram for a BJT amplifier with (b) the small signal response for different bias conditions on the gate.

4.2.3 Gain vs. Bias Condition for a BJT Amplifier

In this example, a BJT amplifier is analyzed for its small signal gain at various bias conditions. Hence, the simulation requires both a DC sweep and an AC small signal perturbation. This dual simulation capability allows a user to analyze how the small signal performance of a device changes with variations in the bias condition.

Figure 4.6 shows the circuit diagram for the BJT amplifier and the small signal gain versus the base bias. V_{in}(DC) is swept from 0.5V to 1.0V in tightly space voltage steps while V_{in}(ac) is applied at a relatively low frequency in order to negate high frequency effects.

The PISCES input deck and the circuit file are given in Figure 4.7. In the PISCES deck, the parameter to specify a joint ac/DC analysis is **acdcckt** on the **solve** card. In the circuit file, the circuit configuration is described in a SPICE-like format and an **.ac** card and a **.dc** card specifies the limits

```
$ find bias for maximum gain in amplifier

$ load mesh generated by bjt.pis
mesh infile=bjt.msh

$ load circuit BC's
contact cktfil=amp.ckt

$ set models
model srh auger bgn conmob fldmob

$ do init solution
symb newton carrier=2
method itlimit=20 trap
solve init
solve v1=0.0 vstep=1.0 nstep=5 elect=1

$ solve for small signal at multiple DC biases
log cktfile=amp.log
solve acdcckt
```

(a)

```
* test for analysis of inverter

* initial biases
vin 12 0 dc 0.5 ac 1.0
vcc 11 0 dc 5.0

* bias circuitry
rin 12 2 50k
rcc 11 1 75k
rf 1 2 10Meg

* numerical device
nq1 1 2 0

* set circuit options
.options area=200 rmin=1e-12 gmin=1e-12

* DC and ac sweep
.dc vin 0.5 1.0 0.005
.ac lin 1 1 1
```

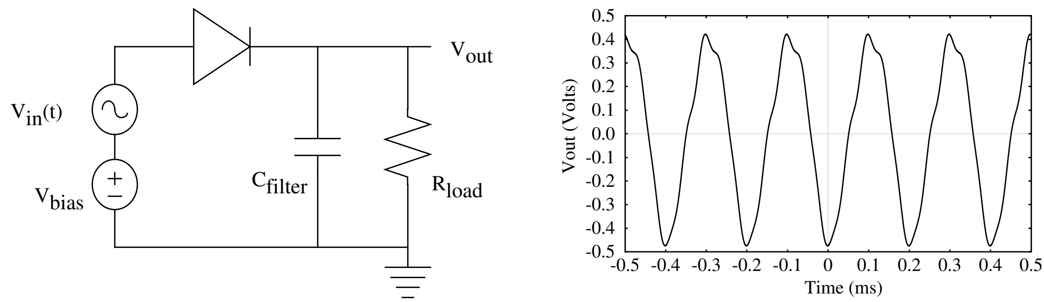
(b)

Figure 4.7 (a) PISCES input deck for determining small signal gain versus bias condition of BJT amplifier. (b) Netlist for circuit and analysis limits.

on the two analyses. The **.ac** card specifies the same start and end frequency with only a single step. When this is encountered on the **.ac** card, the small signal simulation is performed only at that one frequency. Single frequency AC analysis is used because it is the only one desired for this example. The user does have the option to do multiple frequency simulations in an ac/DC simulation.

4.3 Harmonic Balance Analysis

Harmonic balance analysis provides a technique to do large signal sinusoidal analysis in the frequency domain rather that the time domain. The next three examples demonstrate the power of this capability by solving for the large signal response of an AM demodulator, a BJT mixer, and MOS power amplifier. All three involve some simple circuitry in order to obtain the desired response.



(a) (b)

$$V_{in}(t) = \left(1 + \frac{1}{2}\sin(2\pi(5x10^{3})t)\right)\sin(2\pi(5x10^{6})t)$$

Figure 4.8 Circuit diagram of an AM demodulator. A 5KHz signal on a 5MHz carrier is fed into the demodulator.

4.3.1 Large Signal Analysis of an AM Demodulator

A simple AM demodulator, consisting of a diode, resistor, and capacitor, is fed a 5KHz signal on a 15MHz carrier as shown in the circuit diagram in Figure 4.8 (a). The diode rectifies the amplitude modulated input voltage F(t). The output capacitor and resistor, acting as a low pass filter, smooths out the resultant signal. The output, less the DC offset, is shown in Figure 4.8 (b). Note that the output signal is not a pure sine wave and some improvements are needed in the circuit and device in order to optimize the results.

The PISCES input deck and the circuit file are shown in Figure 4.9. The interesting aspect of this simulation is how the input signal is represented. In describing harmonic balance sources (page 35), only a single frequency is specified; hence, the expression for $F(t)$ must be reduced to single sinusoids. This reduction is easily accomplished using trigonometric identities in order to generate Equation 6.1.

$$F(t) = \sin(\omega_c t) + 0.25(\cos((\omega_c - \omega_s)t) - \cos((\omega_c + \omega_s)t)) \tag{6.1}$$

where ω_c is the carrier frequency and ω_s is the signal frequency.

```
$ PISCES simulation deck for am demodulator

$ load mesh generated by diode.pis
mesh in=diode.msh

$ tell PISCES about circuit description
contact cktfile=am.ckt

$ do symbolic factorization
symb newton carr=2

$ set numerical methods for PISCES and HB
method biaspart
hbmeth gmrestart=10 maxiter=40

$ set simulation models
models temp=300 srh auger conmob fldmob

$ do initial solution and operating point solution
solve init
solve v1=0.0 vstep=0.2 nstep=3 elect=1
solve opckt

$ set circuit solution file and do HB
$   analysis to find demodulated signal
log column.out cktfile=am.log
solve hbckt
```

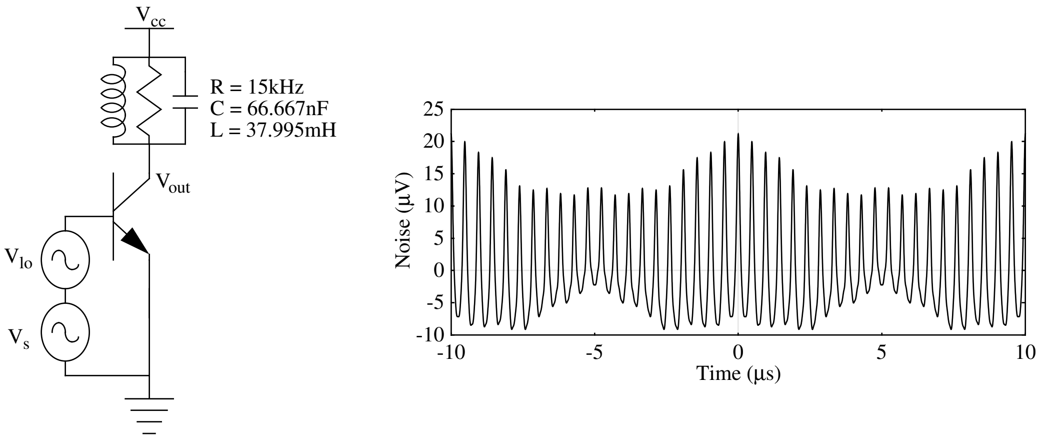
(a)

```
* a simple AM demodulator

* 15MHz carrier plus 5kHz signal
Vcar 1 0 hb 0.65 0.0 15Meg 0
Vsigm 2 1 hb 0.0 0.25 14.995Meg 90
Vsigp 3 2 hb 0.0 -0.25 15.005Meg 90

* rectifier
Ndiode 3 4

* load resistance and filtering capacitance
Cfilter 4 0 0.013u
rload 4 0 5k

* frequencies of interest
.hb 15Meg 5 5k

* ramp up ac carrier signal magnitude for
* improved convergence
.hbac Vcar 0.0 1.0 0.1

* circuit options
.options gmin=0 rmin=0 area=100
```

(b)

Figure 4.9 (a) PISCES input deck for analyzing an AM demodulator. (b) Netlist describing circuit diagram and analysis limits.

Another issue of concern is the convergence. Harmonic balance convergence is based upon the frequency of the simulation and the amount of power or amplitude in the input signals. When either becomes large, convergence may become difficult. In order to facilitate improved convergence, the carrier frequency is ramped up from 0.0V to the 1.0V desired in the simulation by including the **.hbac** card in the circuit netlist. The solutions for the intermediate steps in ramping the carrier frequency are included in the output file, but those results are easily dismissed during analysis.



(a) (b)

Figure 4.10 (a) A BJT Mixer that down converts a 200MHz to a 100KHz signal. (b) The noise in the 100kHz output signal.

4.3.2 BJT Mixer to Down Covert a 2MHz Signal to 100KHz

This example consists of a simple, single-stage BJT mixer using an HP25 transistor from Hewlett Packard as shown in Figure 4.10. The input to the circuit is driven by a 2.1MHz LO and a 2.0MHz RF signal, and mixes the signal down to 100KHz. A tuned resonant *RLC* circuit with a high Q in the collector filters out mixing products other than 100KHz. The noise in the output, also shown in Figure 4.10, is in the micro-voltage range thus providing a very clear output signal. This distortion can be reduced by increasing the Q of the resonator and reducing the RF power level.

The PISCES input file and circuit file are shown in Figure 4.11. In this example, it is very important to ramp up the DC bias on the mixer. This biasing is done in the PISCES file by first increasing the voltage on the collector and then the voltage on the base. Upon obtaining the DC bias the harmonic balance simulation can then be executed with better convergence behavior.

```
title HP25 w/ 3-emitter, 2-base, 1-collector

$ load mesh for structure

$ set finite recombination on emitter
contact num=3 surf.rec vsurfn=1e5 vsurfp=1e5
contact cktfil=hp25mix.ckt

$ do symbolic factorization, set numerical meth-
ods and models
symbol carrier=2 newton
method itlim=50 p.tol=1e-7 c.tol=1e-7
models temp=300 bgn srh auger conmob fldmob

$ initial solve
solve init

$ sweep v1 (collector) up to 10 volts
solve v1=1 nstep=9 elect=1 vstep=1

$ sweep V2 (base) up to 0.7 volts
solve v2=0.1 nstep=5 elect=2 vstep=0.1

$ specify output file and do HB analysis
log cktfile=hp25mix.log column.out
solve hbckt
```

(a)

```
* Single-device BJT mixer based on the HP-25
* Q = 100, LO v=0.15

* bias voltages
Vcc 4 0 10.0
Vlo 1 0 hb 0.7 0.15 2.1Meg 0

* input signal
Vs 2 1 hb 0.0 0.01 2.0Meg 0

* resonant RLC circuit
L1 4 3 37.995e-6
C1 4 3 66.667e-9
R1 4 3 15k

* HP25 bjt
Mbyte 3 2 0

* harmonic balance limits
.hb 2.1Meg   6   0.1Meg

* circuit issues
.options gmin=1e-12 area=5 rmin=1e-12
```

(b)

Figure 4.11 (a) PISCES input deck for analyzing the BJT Mixer. (b) Netlist describing circuit diagram and analysis limits.

4.3.3 Analyzing Gain and Efficiency in an MOS Power Amplifier

The modeling of an RF device coupled with Parasitics, matching network, and bias network can provide insight into the performance of a discrete power amplifier. Figure 4.12 shows the intrinsic LDMOS device with Parasitics at the chip level due to the pad and interconnect. In addition to the parasitics shown, there are additional parasitics for the packaging. The input and output of the amplifier is connected through matching networks that are tuned for the individual device and isolated from the DC bias by a capacitor. A biasing network sets the operating point and is isolated from the

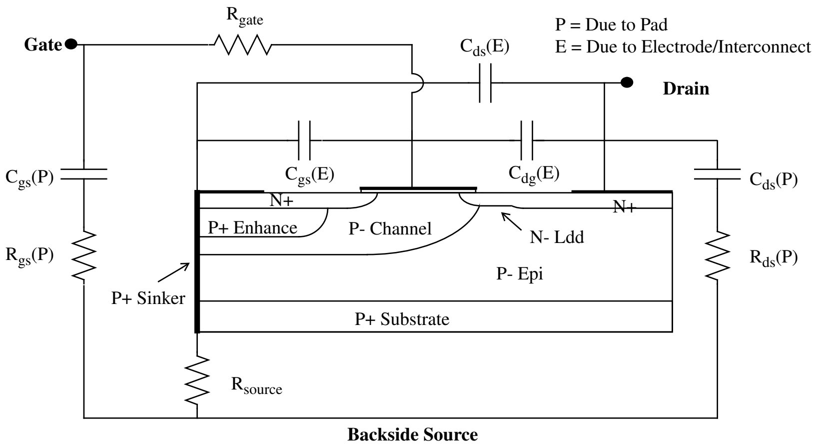


Figure 4.12 Cross section of an LDMOS transistor with parasitics represented by lumped elements.

input by an inductance. A PISCES deck and netlist for the device and circuit is created in a similar manner to the previous examples and thus not provided here.

The physics of the device plays an important role in its performance. A laterally diffused graded channel enhances RF performance, prevents punch-through, and increases the device transconductance. A p+ sinker (represented by a side contact electrode) connects the source and substrate together to eliminate extra bonding wires and provide for a back side contact. An n- LDD decreases the electric field at the drain side of the channel and optimizes R_{ds}(on), BV_{dss}, and C_{dg}. A metal field plate reduces the electric fields at the edge of the gate, thereby increasing the breakdown voltage and reducing C_{dg} [8] [9].

The response of interest is the gain and efficiency as shown in Figure 4.13. The gain rolls off at higher power levels (P_{in} > 30 dBm) because the device operates in g_m compression region and the output power is limited by the saturation current. The power added efficiency (PAE) is low for a small P_{out} because the device drains more power in Class A operation. Efficiency increases until just after the gain starts to decrease. At this point, P_{in} approaches P_{out} resulting in very little power added to the input signal.

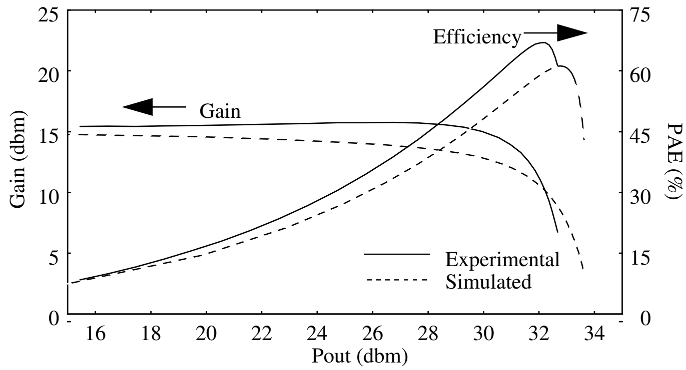


Figure 4.13 Gain and power added efficiency (PAE) for LDMOS power amplifier.

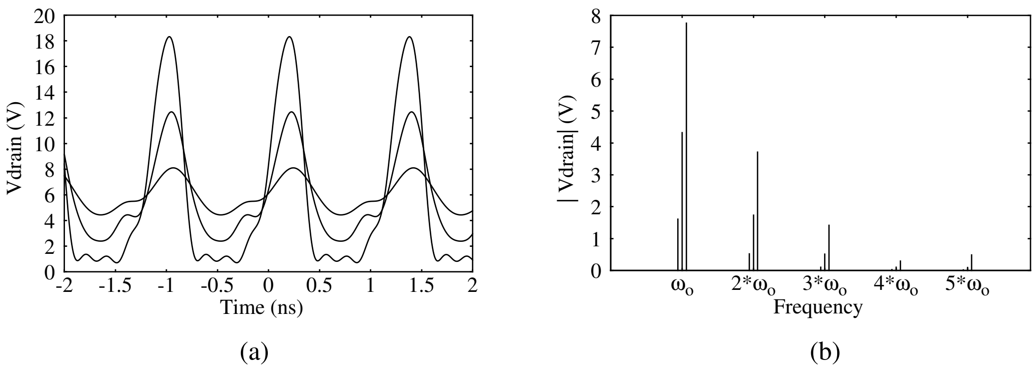


Figure 4.14 (a) Time domain voltage and (b) spectrum of the voltage on the drain of the LDMOS transistor in a power amplifier as the device enters g_m compression.

In addition to examining a calculated result of the simulation, one can analyze the state of the device at various levels in the simulation. For example, the device performance begins to degrade when it enters g_m compression. Figure 4.14 shows the voltage on the drain in the time domain and frequency domain as the device enters this region. The three power levels correspond to an output of 19.9, 28.5, and 33.5 dBm. At the low power level the voltage on the drain is able to swing to its full limits. As the power increases, the voltage becomes limited on how low it can swing. The time domain plot and the frequency plot reflect that change by two different mechanisms. The time domain signal flattens out when it swings low. The spectrum reflects this effect with the addition of harmonics at the higher frequencies.

This example clearly shows the power of the harmonic balance device simulation. The physical characteristics of an advanced technology impact simulation significantly. Hence, this tool allows the design engineer to analyze and optimize a robust devices in order to achieve the maximum performance [10].

References

[1] B. Troyanovsky, *Frequency Domain Algorithms for Simulating Large Signal Distortion in Semiconductor Devices,* dissertation, Department of Electrical Engineering, Stanford University, November 1997.

[2] F. Rotella, *Mixed Circuit and Device Simulation for Analysis and Design of RF Components*, dissertation to be Submitted to Department of Electrical Engineering, Stanford University.

[3] B. Troyanovsky, F. Rotella, Z. Yu, R. Dutton, and J. Sato-Iwanga, "Large Signal Analysis of RF/Microwave Devices with Parasitics Using Harmonic Balance Device Simulation," SASIMI, Fukuoka, Japan, Nov. 1996.

[4] B. Johnson, T. Quarles, A. R. Newton, D.O Pederson, and A. Sangiovanni- Vincentelli, SPICE3 Version 3f User's Manual, Berkeley: Regents of the University of California, 1992.

[5] W. J. McCalla, *Fundamentals of Computer-Aided Circuit Simulation*, Boston: Kluwer Academic Publishers, 1993.

[6] S. E. Laux, "Techniques for small-signal analysis of semiconductor devices," IEEE Transactions on Electronic Devices, Vol. ED-32, No. 10, pp. 22028-2037, 1985.

[7] R. Bank, W. M. Coughran, W. Fichtner, E. H. Grosse, D. J. Rose, and R. K. Smith, "Transient simulation of silicon devices and circuits," IEEE Transactions on Electron Devices, pp. 1992-2007, .Oct. 1985.

[8] G. Ma, W. Burger, C. Dragon, and T. Gillenwater, "High Efficiency LDMOS Power FET for Low Voltage Wireless Communications," Proceedings of IEDM. San Francisco, CA, December 1996.

[9] A. Wood, C. Dragon, and W. Burger, "High Performance Silicon LDMOS Technology for 2GHz RF Power Amplifier Applications," Proceeding of IEDM. San Francisco, CA, December 1996.

[10] F. Rotella, G. Ma, Z. Yu, R. Dutton, "Modeling, Analysis, and Design of RF LDMOS Devices Using Harmonic Balance Device Simulation," Paper submitted to IEEE Transactions on Microwave Theory and Techniques, January 1999.