IEEE Int. Conference on Computer-Aided Design (ICCAD'84); Santa Clara, CA, November 12–15, 1984, pp.A.1–4.

INTERFACING CIRCUIT SIMULATION WITH OPTIMIZATION PACKAGES

W.M. Zuberek

Department of Computer Science, Memorial University
St. John's, NL, Canada A1C–5S7

Abstract: The most effective optimization techniques assume that the objective (and constraint) functions are differentiable, and their (at least) first derivatives are available. In the case of circuit simulation, even if the derivatives can be obtained analytically, the amount of calculations can be quite excessive, and then numerical approximation of gradient information can be the simplest and the most effective solution. The paper describes an interface between a nongradient circuit simulator and gradient optimization packages.

1. INTRODUCTION

Computer-aided circuit analysis, or circuit simulation, has become a widely accepted tool in the area of integrated circuit design. By using this method, the circuit designer can easily investigate the effects of different designs for a given circuit topology. The process of selecting an appropriate design to satisfy the required specifications on the circuit performance is usually based on consecutive approximations as well as on the designer's experience, and there is a clear need for efficient optimization techniques which could be applied to the nominal design problem [1,2,8].

Despite considerable research activity in optimization of electronic circuits, optimization techniques have not been used as widely as might be expected. The reasons which are most frequently named to explain this situation are [7,8]: the difficulty of linking optimization packages with simulation programs, the inadequacy of the optimization algorithms used, and the lack of familiarity with these tools.

An example of an optimization system which overcomes some of these difficulties is the DELIGHT.SPICE system [7], the result of the union of the DELIGHT interactive optimization-based system and the SPICE circuit simulator, both developed at the University of California, Berkeley. However, sophistication of problem descriptions, evaluation of gradients by finite differences, and inefficient interface between the DELIGHT and SPICE programs are indicated [7] as primary areas where further research is needed.

The paper describes an interface between a nongradient circuit simulation package (SPICE-PAC version 2G6a) and general gradient optimization packages. The interfacing routines provide internal scaling, lower and upper bounds on optimization variables, and numerical approximation of gradients which are required by the most efficient optimization algorithms. An example of simulation and optimization is included as an illustration of interfacing.

2. CIRCUIT SIMULATION

A flexible and efficient organization of repeated simulations of circuits with fixed topology cannot be provided by traditional simulation programs which are usually batch-oriented, and which require a new, independent run in the case of even a minor change in any of the element descriptions or parameter values. A new structure of circuit simulators is needed in which different analyses can be performed selectively, in which there is an access to internal representation of circuit elements in order to modify their values, and which can easily be interfaced with other computer-aided design tools, for example optimization techniques, device and process simulation, e.t.c. The simulators should have the structure of a set (or a package) of subroutines rather than a program with one, fixed sequence of operations.

SPICE-PAC version 2G6a [11,13] is a package of simulation subroutines obtained by redesigning the SPICE 2G.6 simulation program [8,9,10]. The package provides: (a) all the analyses available in the SPICE 2G programs, (b) a hierarchical naming scheme for subcircuits, (c) an access to circuit variables as required in circuit optimization, (d) dynamic definitions of parameters for all analyses, (e) dynamic declarations of outputs, (f) parameterized subcircuit calls, and (g) an interface to hierarchical libraries of standard modules (or "building blocks").

SPICE-PAC contains 25 main subroutines [11] (SPICEA to SPICEY) but does not provide the "main" program which must be supplied by the user to "drive" the subroutines, i.e., to call the subroutines which read the circuit description, define parameters and perform analyses, as required by a particular application.

SPICE-PAC, similarly as the SPICE 2G.6 program, does not provide the gradient information.

3. OPTIMIZATION ROUTINES

The nominal design problem is usually transformed to a multidimensional optimization problem. It is rarely the case that a single objective has to be optimized. In most applications, various objectives compete against each other and a compromise has to be reached. Combining several objectives into a single cost function has the disadvantage of hiding the physical significance of these objectives, particularly acute in an interactive environment. Also, circuit specifications and performance requirements are, in general, functions of the circuit variables (or design parameters) through circuit responses, and objective (and constraint) function evaluations require the solution of very large sets of simultaneous nonlinear equations. Consequently, the evaluations are inaccurate but expensive.

On the other hand, the development of quality optimization software is a difficult task which requires research in many areas [3,5]. Many of the available optimization packages are inadequate because the number of test problems is too small or the starting points are too close to the solution; in addition, quite often there has been too much emphasis on measuring the efficiency and not enough on testing reliability and robustness [6].

The choice of optimization algorithms depends not only on the type of problems to be solved, but also on the available information about the cost of evaluating the functions, the dimensions of the problem, and the performance of the algorithm during extensive testing on problems that are believed to be typical [3,6]. In many cases constrained minimax optimization techniques are preferred due to a rather simple handling of multiple objectives, and efficiency and reliability at the same time [2].

The general structure of interfacing SPICE-PAC with an (abstract) optimization package is shown in Fig.1. The main segment, MAIN, initializes the simulation package (using the subroutines SPICEA, SPICEB and SPICEC) and calls the optimization package OPTIM-PAC indicating the subroutine SUBR as one of parameters (usually the first one). SUBR is a user defined subroutine which evaluates objective (and constrained) functions (using the simulation package) for circuit variables determined by OPTIM-PAC. Whenever the optimization package requires an evaluation of objective (and constraint) functions, it simply calls the subroutine indicated in the calling sequence. Such "indirect communication" is typical for optimization software and there are only few packages which use "reverse communication" instead [3,5].

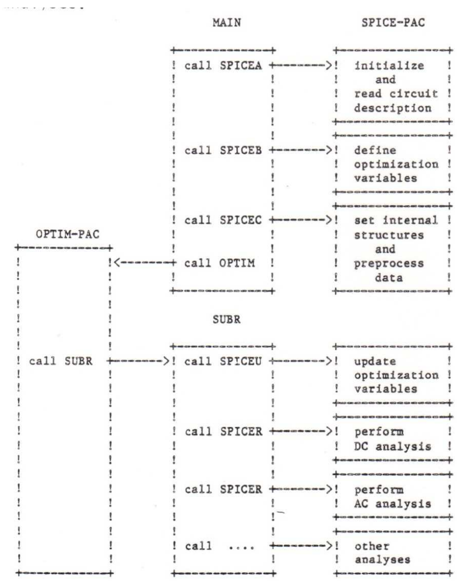

Fig.1. Interfacing SPICE-PAC with an abstract optimization package.

4. INDIRECT COMMUNICATION

The general idea of indirect communication can be extended to multi-level indirection which can be very useful if additional subroutines are to be used as an interface between optimization and simulation packages. Since the SPICE-PAC package does not provide the gradient information which is required by practically all modern optimization packages, the simplest solution is to insert numerical approximation of gradients between the gradient optimization package (OPTIM-PAC in Fig.1) and the subroutine SUBR evaluating objective (and constraint) functions. A modified Broyden method can be used for numerical approximation of gradients [12].

The optimization package WMBG2 used in the following example is in fact an extension of the linearly constrained minimax optimization technique due to Hald [4] (the WMLC2 package) combined with the WGRD2 package for numerical approximation of gradients [12], as shown in Fig.2. The package provides internal scaling, lower and upper bounds on optimization variables, and several entries which correspond to different types of user-defined subroutines SUBR.

```
 OPTIM-PAC
+---------+
|         |<--- MAIN
|  WMBG2  |
|         |   +---------+   +---------+   +---------+
|         |-->|         |-->|         |-->|         |--> SUBR
|         |   |  WMLC2  |   |  WGRD2  |   |         |
|         |   |         |   |         |   |         |
|         |   +---------+   +---------+   |         |
|         +-------------------------------+         |
|                                                   |
+---------------------------------------------------+
```

Fig.2. General structure of the WMBG2 package.

5. OPTIMIZATION EXAMPLE

As an optimization example a simple single-stage

CE amplifier in a self-biasing configuration is analyzed, and it is to find the values of resistors R1, R2 and RE such that for the midband frequency f=50KHz, for beta.dc=80, 150, 250, and for the temperatures T=-50, 27 and 100 degrees Celsius, the magnitude of the voltage gain is equal to 10 V/V and the input resistance is not less than 10Kohms.

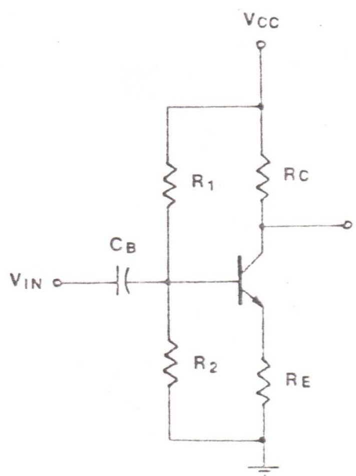

Fig.3. Optimization example.

In minimax formulation, the optimization variables are R1, R2 and RE, and the 19 residual functions are:

- the difference between 10K and the minimum input resistance (decreased 1000 times),
- the differences between the magnitude of the voltage gain and 10 V/V for the temperatures T=-50, 27, 100 degrees C and for beta.dc=80, 150, 250,
- the differences between 10 V/V and the magnitude of the voltage gain for the temperatures T=-50, 27, 100 degrees C and for beta.dc=80, 150, 250.

```
****  SPICE-PAC 2G6a.84.05  DATE : 15.95.84 15:52
****  INPUT LISTING         TEMP =   27.000 DEG C
***************************
* AMPLIFIER OPTIMIZATION *
***************************
VCC 5 0 12
VIN 1 0 AC 1
R1 2 5 100K
R2 2 0 10K
RC 4 5 5K
RE 3 0 300
CB 1 2 100UF
Q1 4 2 3 MOD
.MODEL MOD NPN(BF=50 VAF=50 IS=1.E-9 RB=100 CJC=1PF)
.PRINT AC V(4) V(2) I(VIN)
.AC 50K
.END/EXT
.VAR R1
.VAR R2
.VAR RE
.PAR/1 TEMP(-50.0)
.PAR/2 TEMP(27.0)
.PAR/3 TEMP(100.0)
.END

VARIABLES :

        R1       R2       RE

STARTING POINT :

     1.00d+05 1.00d+04 3.00d+02

LOWER AND UPPER BOUNDS :

     1.00d+04 5.00d+03 1.00d+02
     5.00d+05 1.00d+05 5.00d+02

ITERATIONS :

        R1       R2       RE     maxfun

  1  1.00d+05 1.00d+04 3.00d+02 6.91d+01
  2  1.00d+05 1.00d+04 3.00d+02 6.91d+01
  3  1.00d+05 1.00d+04 3.00d+02 6.91d+01
  4  1.00d+05 1.00d+04 3.00d+02 6.91d+01
  5  1.21d+05 9.58d+03 3.98d+02 4.62d+01
  6  1.96d+05 1.49d+04 4.59d+02 2.24d+01
  7  2.08d+05 1.42d+04 4.46d+02 2.38d+01
  8  2.86d+05 2.32d+04 4.74d+02 6.01d-01
  9  4.01d+05 2.89d+04 4.25d+02 5.14d-01
 10  4.47d+05 3.54d+04 4.45d+02 2.69d-01
 11  4.47d+05 3.54d+04 4.45d+02 2.66d-01
 12  2.65d+05 1.89d+04 4.54d+02 1.02d+01
 13  4.03d+05 3.08d+04 4.39d+02 2.20d-01
 14  3.55d+05 3.25d+04 4.51d+02 7.35d-02
 15  3.55d+05 3.24d+04 4.51d+02 6.66d-02
 16  3.45d+05 3.34d+04 4.52d+02 2.17d-02
 17  3.39d+05 3.38d+04 4.53d+02 4.63d-03
 18  3.38d+05 3.38d+04 4.53d+02 3.25d-03
 19  3.38d+05 3.38d+04 4.53d+02 4.22d-03
 20  3.38d+05 3.38d+04 4.53d+02 3.36d-03

SOLUTION :

     3.38d+05 3.38d+04 4.53d+02

     NUMBER OF ITERATIONS :  13
     NUMBER OF SHIFTS :       1
```

The solution is obtained in 13 iteration steps with 20 function evaluations and the maximum residual function at the solution is less than 0.004 . It can be observed that the solution is in a rather narrow and quite flat valley. In many cases a less accurate solution, obtained in just a few iteration steps, should be satisfactory.

6. CONCLUDING REMARKS

Interactive simulation and optimization systems, designed to permit the user to interrupt, diagnose, modify and restart a computation as it progresses, can stimulate new design methodologies but require software tools which are much more flexible, reliable and general than the existing ones. Also, man-machine environments must be designed in such a way that non-expert users can easily access expert tools and systems. For this to occur, further research in many areas is necessary.

REFERENCES

[1] K.J. Antreich, S.A. Huss, "An interactive optimization technique for the nominal design of integrated circuits"; IEEE tr. Circuits and Systems, vol.31, no.2, pp.203-211, 1984.

[2] R.K. Brayton, G.D. Hachtel, A.L. Sangiovanni-Vincentelli, "A survey of optimization techniques for integrated-circuit design"; Proc. of the IEEE, vol.69, no.10, pp.1334-1362, 1981.

[3] P.E. Gill, W. Murray, S.M. Picken, M.H. Wright, "The design and structure of a Fortran program library for optimization"; ACM Trans. Mathematical Software, vol.5, no.3, pp.259-283, 1979.

[4] J. Hald, "MMLA1Q, a Fortran subroutine for linearly constrained minimax optimization"; Inst. for Numerical Analysis, Technical University of Denmark, Lyngby, Denmark, Report NI-81-01, 1981.

[5] J.J. More, "Notes on optimization software"; NATO Advanced Research Institute on Nonlinear Optimization, Cambridge, England, 1981.

[6] J.J. More, B.S. Garbow, K.E. Hillstrom, "Testing unconstrained optimization software"; ACM tr. Mathematical Software, vol.7, no.1, pp.17-41, 1981.

[7] B. Nye, A. Sangiovanni-Vincentelli, J. Spoto, A. Tits, "DELIGHT.SPICE, an optimization-based system for the design of integrated circuits"; Proc. 1983 Custom Integrated Circuits Conf., Rochester NY, 1983.

[8] A.R. Newton, D.O. Pederson, A.L. Sangiovanni-Vincentelli, C.H. Sequin, "Design aids for VLSI - The Berkeley perspective"; IEEE Trans. Circuit and Systems, vol.28, no.7, pp.666-680, 1981.

[9] D.O. Pederson, "A historical review of circuit simulation"; IEEE Trans. Circuits and Systems, vol.31, no.1, pp.103-111, 1984.

[10] A. Vladimirescu, K. Zhang, A.R. Newton, D.O. Pederson, A.L. Sangiovanni- Vincentelli, "SPICE Version 2G - User's Guide (10 Aug. 1981)"; Department of Electrical Engineering and Computer Sciences, University of California, Berkeley CA 94720, 1981.

[11] W.M. Zuberek, "SPICE-PAC 2G6a.84.05 - User's Guide"; Department of Computer Science, Memorial University of Newfoundland, St. John's, Nfld., Canada A1C 5S7, Technical Report #8404, 1984.

[12] W.M. Zuberek, "Numerical approximation of gradients for circuit optimization"; Proc. 27 Midwest Symp. on Circuits and Systems, Morgantown WV, pp.200-203, 1984.

[13] W.M. Zuberek, "SPICE-PAC, a package of subroutines for interactive simulation and optimization of circuits"; Proc. IEEE Int. Conf. on Computer Design, Port Chester NY, pp.492-496, 1984.