27-th Midwest Symposium on Circuits and Systems (MSCAS'84), Morgantown, WV, June 11–12, 1984, pp.200-203.

NUMERICAL APPROXIMATION OF GRADIENTS FOR CIRCUIT OPTIMIZATION

W.M. Zuberek

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

Abstract. An implementation of a modified Broyden method for approximation of gradients is described and some simple comparisons of the original and modified approximations are given. An example of circuit optimization is included in which gradient approximation routines are used as an interface between a nongradient circuit simulator and a gradient optimization technique.

1. INTRODUCTION

Many numerical methods require repeated evaluation of the first derivatives of given functions of several variables. In optimization, the most effective algorithms assume that the objective (and constraint) functions are differentiable, and that the functions and their Jacobian matrices can be evaluated (usually by user-defined subroutines) at consecutive points determined by the optimization algorithm. Quite often, however, evaluating the Jacobian matrix can be rather difficult from the point of view of practical calculations [1]. Even if the (objective and constraint) functions are sufficiently simple for their partial derivatives to be obtained analytically, the amount of calculations to evaluate all of them can be quite excessive. In the case of circuit optimization, when objective functions are evaluated by circuit simulations (e.g., in the time domain), numerical approximation of the gradient information can be the simplest and the most effective solution. This can be done directly by differencing function values, i.e., by evaluating differences of function values that correspond to small perturbations of independent variables (one at a time), or indirectly, by updating the initial approximation of the Jacobian matrix in consecutive iteration steps (for example using Broyden formula [2,3]). For n independent variables, the direct approach requires (n+1) function evaluations per approximation, while in indirect methods the average number of function evaluations per approximation is usually bewteen 1 and 2.

The paper describes an implementation of a modified Broyden method in which the initial approximation of the Jacobian matrix is obtained by direct differencing function values, and then the updating steps require 1.33 function evaluation on average. An example of simulation and optimization of a simple circuit is included as an illustration of interfacing a nongradient circuit simulator (the SPICE-PAC package [7]) with a gradient optimization technique (linearly constrained minimax optimization [4]).

2. BROYDEN FORMULA

For a set of m (nonlinear) functions $f = [f_1, ..., f_m]^T$ of n variables $x = [x_1, ..., x_n]^T$, the Broyden formula [2,3,6] for approximations B_k of the Jacobian matrices J_k at points x_k, $k = 1, 2, ...$, is given by

$$B_k = B_{k-1} + (d_k - B_{k-1}h_k{h_k}^T/({h_k}^T h_k)$$

where $d_k = f(x_k) - f(x_{k-1})$, $h_k = x_k - x_{k-1}$, and the initial approximation B_0 is usually obtained by direct differencing

$$b_{i,j} = (f(x_0 + \delta e_j) - f(x_0))/\delta; \quad i = 1, ..., m; j = 1, ..., n$$

where δ is a small positive number, and e_j is the normalized j-th coordinate vector.

The main advantage of the Broyden formula is that it does not require additional function evaluations, and consecutive approximations can be evaluated at points x_k determined by an optimization algorithm. It can be observed, however, that in some cases the approximations can be quite inaccurate.

(1) If consecutive points x_k are selected in such a way that (at least) one of variables, say x_ℓ, is constant, the corresponding elements of increments h_k are zero, and consecutive approximations of partial derivatives with respect to x_ℓ do not change.

(2) If linear predictions based on approximated Jacobians do not match the function changes (i.e., if $(d_k - B_{k-1}h_k)$ is nonzero), the corrections of partial derivatives are proportional to components of the increment h_k. In particular, if there are functions which are linear with respect to some variables, and if the corresponding elements of h_k are nonzero, then the approximations of (constant) partial derivatives are updated by nonzero values.

(3) If consecutive steps $x_k, k = 1, 2, ...$, are collinear, the effects of (1) and (2) can cumulate and deteriorate the approximations B_k even if the sequence x_k converges to a point x_s.

The observations are illustrated by a simple example in which the gradient of a single function

$$f = {x_1}^2 x_2 + x_3$$

is approximated at points

$$x_0 = [1, 2, 1]^T, x_k = 0.5(x_{k-1} + [2, 2, 2]^T); \; k = 1, ..., 15.$$

For each k, the first line shows the function value $f(x_k)$ and the variables x_k, while the second line contains the second norm of the difference between the gradient and its approximation, norm2$(J_k - B_k)$, and errors of approximated partial derivatives, $J_k - B_k$.

0	3.000d+00	1.000d+00	2.000d+00	1.000d+00
	2.000d-05	-2.000d-05	-1.000d-12	-1.000d-12
1	6.000d+00	1.500d+00	2.000d+00	1.500d+00
	2.016d+00	1.500d+00	1.250d+00	-5.000d-01
2	7.875d+00	1.750d+00	2.000d+00	1.750d+00
	2.980d+00	1.750d+00	2.062d+00	-1.250d+00
3	8.906d+00	1.875d+00	2.000d+00	1.875d+00
	3.533d+00	1.875d+00	2.516d+00	-1.625d+00
4	9.445d+00	1.938d+00	2.000d+00	1.938d+00
	3.824d+00	1.937d+00	2.754d+00	-1.812d+00
5	9.721d+00	1.969d+00	2.000d+00	1.969d+00
	3.973d+00	1.969d+00	2.876d+00	-1.906d+00
6	9.860d+00	1.984d+00	2.000d+00	1.984d+00
	4.048d+00	1.984d+00	2.938d+00	-1.953d+00
7	9.930d+00	1.992d+00	2.000d+00	1.992d+00
	4.085d+00	1.992d+00	2.969d+00	-1.977d+00
8	9.965d+00	1.996d+00	2.000d+00	1.996d+00
	4.104d+00	1.996d+00	2.984d+00	-1.988d+00
9	9.982d+00	1.998d+00	2.000d+00	1.998d+00
	4.114d+00	1.998d+00	2.992d+00	-1.994d+00
10	9.991d+00	1.999d+00	2.000d+00	1.999d+00
	4.118d+00	1.999d+00	2.996d+00	-1.997d+00
11	9.996d+00	2.000d+00	2.000d+00	2.000d+00
	4.121d+00	2.000d+00	2.998d+00	-1.999d+00
12	9.998d+00	2.000d+00	2.000d+00	2.000d+00
	4.122d+00	2.000d+00	2.999d+00	-1.999d+00
13	9.999d+00	2.000d+00	2.000d+00	2.000d+00
	4.123d+00	2.000d+00	3.000d+00	-2.000d+00
14	9.999d+00	2.000d+00	2.000d+00	2.000d+00
	4.123d+00	2.000d+00	3.000d+00	-2.000d+00
15	1.000d+01	2.000d+00	2.000d+00	2.000d+00
	4.123d+00	2.000d+00	3.000d+00	-2.000d+00

3. CORRECTION STEPS

To overcome some deficiencies of the Broyden formula, and in particular to eliminate sequences of linearly dependent consecutive steps, the so called "correction steps" have been proposed [5] which are based on the method of linear independent directions of Powell [6]. Before every q-th "normal" approximation (q is a parameter), an additional approximation is performed with an increment h chosen in such a way that the "uniform linear independence" of directions is satisfied.

For the previous example, the modified approximations of gradients (with the parameter q equal to 2) are as follows:

0	3.000d+00	1.000d+00	2.000d+00	1.000d+00
	2.000d-05	-2.000d-05	-1.000d-12	-1.000d-12
1	6.000d+00	1.500d+00	2.000d+00	1.500d+00
	2.016d+00	1.500d+00	1.250d+00	-5.000d-01
2	7.875d+00	1.750d+00	2.000d+00	1.750d+00
	2.425d+00	1.250d+00	2.062d+00	2.500d-01
3	8.906d+00	1.875d+00	2.000d+00	1.875d+00
	2.736d+00	8.750d-01	2.516d+00	-6.250d-01
4	9.445d+00	1.938d+00	2.000d+00	1.938d+00
	1.261d+00	1.031d+00	1.057d-01	-7.188d-01
5	9.721d+00	1.969d+00	2.000d+00	1.969d+00
	1.346d+00	9.688d-01	2.278d-01	-9.063d-01
6	9.860d+00	1.984d+00	2.000d+00	1.984d+00
	3.003d-01	7.812d-02	2.895d-01	1.563d-02
7	9.930d+00	1.992d+00	2.000d+00	1.992d+00
	3.276d-01	5.469d-02	3.206d-01	-3.906d-02
8	9.965d+00	1.996d+00	2.000d+00	1.996d+00
	7.890d-02	6.445d-02	7.293d-03	-4.492d-02
9	9.982d+00	1.998d+00	2.000d+00	1.998d+00
	8.427d-02	6.055d-02	1.509d-02	-5.664d-02
10	9.991d+00	1.999d+00	2.000d+00	1.999d+00
	1.964d-02	4.883d-03	1.900d-02	9.766d-04
11	9.996d+00	2.000d+00	2.000d+00	2.000d+00
	2.137d-02	3.418d-03	2.095d-02	-2.441d-03
12	9.998d+00	2.000d+00	2.000d+00	2.000d+00
	4.932d-03	4.028d-03	4.585d-04	-2.808d-03
13	9.999d+00	2.000d+00	2.000d+00	2.000d+00
	5.268d-03	3.784d-03	9.467d-04	-3.540d-03
14	9.999d+00	2.000d+00	2.000d+00	2.000d+00
	1.231d-03	3.052d-04	1.191d-03	6.104d-05
15	1.000d+01	2.000d+00	2.000d+00	2.000d+00
	1.339d-03	2.136d-04	1.313d-03	-1.526d-04

The influence of the parameter q is shown in another example in which the Jacobian matrix of a set of 3 functions

$$f_1 = {x_1}^2 + {x_2}^2 + {x_3}^2,\, f_2 = x_1 x_2 + x_3,\, f_3 = {x_1}^2 + {x_2}^3 + x_3.$$

is approximated at points

$$x_0 = [2,2,2]^T,\, x_k = 0.5(x_{k-1} + [-1,-2,-0.5]^T);\; k = 1,...,15$$

and the second norms of the differences between exact and approximated gradients are shown for q = 0,1,2 and 3, where $q = 0$ corresponds to the original Broyden method, without correction steps.

FUNCTION : F1 = X1*X1 + X2*X2 + X3*X3

	0	1	2	3
0	3.46d-05	3.46d-05	3.46d-05	3.46d-05
1	2.80d+00	3.64d+00	2.80d+00	2.80d+00
2	1.40d+00	2.20d+00	1.90d+00	1.40d+00
3	6.99d-01	2.46d+00	1.02d+00	1.42d+00
4	3.49d-01	1.80d+00	1.21d+00	6.13d-01
5	1.75d-01	9.35d-01	7.66d-01	5.33d-01
6	8.73d-02	6.19d-01	2.64d-01	5.58d-01
7	4.37d-02	2.76d-01	1.17d-01	5.06d-01
8	2.18d-02	1.88d-01	6.55d-02	5.05d-01
9	1.09d-02	9.02d-02	2.86d-02	4.00d-02
10	5.46d-03	4.70d-02	1.65d-02	2.70d-02
11	2.73d-03	2.51d-02	7.29d-03	2.66d-02
12	1.36d-03	1.23d-02	4.10d-03	4.95d-03
13	6.82d-04	6.49d-03	1.79d-03	3.30d-03
14	3.41d-04	3.40d-03	1.03d-03	3.25d-03
15	1.71d-04	1.66d-03	4.55d-04	6.26d-04

FUNCTION : F2 = X1 * X2 + X3

	0	1	2	3
0	3.00d-12	3.00d-12	3.00d-12	3.00d-12
1	1.67d+00	2.67d+00	1.67d+00	1.67d+00
2	2.00d+00	2.00d+00	1.37d+00	2.00d+00
3	2.26d+00	1.85d+00	1.42d+00	1.68d+00

4	2.41d+00	8.59d-01	4.68d-01	1.76d+00
5	2.48d+00	2.82d-01	3.45d-01	1.83d+00
6	2.52d+00	3.24d-01	1.12d-01	1.02d+00
7	2.54d+00	8.78d-02	4.71d-02	1.03d+00
8	2.55d+00	3.22d-02	4.18d-02	1.04d+00
9	2.56d+00	3.88d-02	3.09d-02	1.92d-02
10	2.56d+00	2.33d-02	9.62d-03	1.73d-02
11	2.56d+00	3.73d-03	8.63d-03	1.81d-02
12	2.56d+00	4.29d-03	9.14d-03	1.86d-02
13	2.56d+00	3.81d-03	9.27d-03	1.88d-02
14	2.56d+00	5.08d-04	6.01d-04	1.89d-02
15	2.56d+00	5.20d-04	5.39d-04	3.01d-04

FUNCTION : F3 = X1*X1 + X2*X2*X2 + X3

	0	1	2	3
0	1.22d-04	1.22d-04	1.22d-04	1.22d-04
1	7.90d+00	8.90d+00	7.90d+00	7.90d+00
2	4.89d+00	1.98d+00	5.22d+00	4.89d+00
3	3.66d+00	3.51d+00	5.29d+00	3.59d+00
4	3.86d+00	5.71d+00	6.13d+00	3.23d+00
5	4.34d+00	5.26d+00	6.72d+00	3.64d+00
6	4.67d+00	1.32d+00	2.12d+00	3.99d+00
7	4.86d+00	1.54d+00	2.29d+00	4.16d+00
8	4.96d+00	7.86d-01	1.01d+00	4.28d+00
9	5.01d+00	2.62d-01	1.06d+00	5.57d-01
10	5.04d+00	2.51d-01	1.34d-01	5.77d-01
11	5.05d+00	1.07d-01	1.41d-01	5.91d-01
12	5.06d+00	3.23d-02	2.91d-02	6.14d-02
13	5.06d+00	2.88d-02	3.10d-02	6.41d-02
14	5.06d+00	1.60d-02	8.39d-03	6.59d-02
15	5.06d+00	3.56d-03	8.84d-03	8.83d-03

It can be observed that the results corresponding to q = 1,2 and 3 are not signifficantly different, and that in all these cases the final approximation is rather good.

4. PRACTICAL IMPLEMENTATION

The modified Broyden method (with the default value of parameter q equal to 3) has been implemented in double precision arithmetic on a VAX-11/UNIX system as a package WGRD2. The package contains several entries (WGRD21, ..., WGRD28) which provide direct interfaces to optimization packages. A typical calling sequence is

```
CALL WGRD21 (SUBR,N,M,X,F,D,IT,W,LW,MD)
```

where:

SUBR is the name of a subroutine which evaluates the (objective or residual) functions; it must be defined as

```
SUBROUTINE SUBR (N,M,X,F,IND)
DOUBLE PRECISION X(N),F(M)
```

N is the number of variables,

M is the number of functions,

X is the vector of variables,

F is the vector of functions (evaluated by SUBR at point X),

D is a matrix which returns the approximated Jacobian,

IT is the Jacobian transpose indicator,

W is the workspace for the package,

LW is the length of the workspace W; it must be at least

$$M + 3N + N * N + M * N + 4,$$

MD is a flag which must be set to zero at the first call (and then the initial approximation is performed by direct differencing) and should be positive in subsequent calls.

Moreover, the entry

```
CALL WGRD20 (L,W,LW)
```

redefines the parameter q which controls correction steps.

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 R1, R2 and RE such that for the midband frequency f=50KHz, 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.

Circuit simulation is provided by the SPICE-PAC package of simulation subroutines [7], and the general structure of interfacing SPICE-PAC with an (abstract) optimization package is shown in Fig.1. The optimization package WMBG2 used in this 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, as shown in Fig.2.

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

- the difference between 10K and the input resistance if input resistance is less than 10K, otherwise zero,
- the differences between the magnitude of the voltage gain and 10 V/V for the temperatures T=-50, 27, 100 degrees C,
- the differences between 10 V/V and the magnitude of the voltage gain for the temperatures T=-50, 27, 100 degrees C.

```
**  SPICE-PAC 2G6a.84.05  DATE : 15.05.84 AT 15:52
**  INPUT LISTING         TEMP =      27.000 DEG C
***************************
* AMPLIFIER OPTIMIZATION *
```

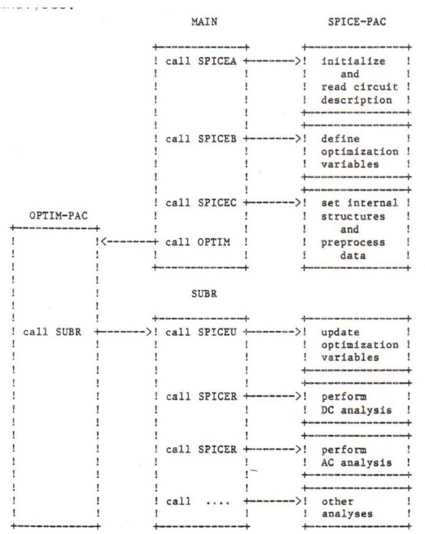

Fig.1. Interfacing SPICE-PAC with an optimization package.

```
***************************
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, which is quite far from the starting point, is obtained in 13 iteration steps with 20 function evaluations, and the maximum residual function at the solution is less than 0.004.

REFERENCES

1. R.K. Brayton, G.D. Hachtel, A.L. Sangiovanni-Vincentelli, "A survey of optimization techniques for integrated-circuit design"; Proc. of the IEEE 1981(69)10, pp.1334-1362.
2. C.G. Broyden, "A class of methods for solving nonlinear simultaneous equations", Mathematics of Computation 1965(19), pp.577-593.
3. C.G. Broyden, "Quasi-Newton methods"; in: Numerical methods for unconstrained optimization, Academic Press 1972, pp.87-106.
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. K. Madsen, "Minimax solution of nonlinear equations without calculating derivatives"; Mathematical Programming Study 3, 1975, pp.110-126.
6. M.J.D. Powell, "Problems related to unconstrained optimization"; in: Numerical methods for unconstrained optimization, Academic Press 1972, pp.29-55.
7. W.M. Zuberek, "SPICE-PAC 2G6a.84.05 - User's Guide"; Department of Computer Science, Memorial University of Newfoundland, St. John's, Canada, 1984.