SIMULATION OPTIMIZATION SYSTEMS
Research Laboratory

KMOS – A FORTRAN LIBRARY FOR NONLINEAR OPTIMIZATION

J.W. Bandler, S.H. Chen and M.L. Renault

SOS-87-1-R

February 1987

McMASTER UNIVERSITY
Hamilton, Canada L8S 4L7
Department of Electrical and Computer Engineering

KMOS – A FORTRAN LIBRARY FOR NONLINEAR OPTIMIZATION

J.W. Bandler, S.H. Chen and M.L. Renault

SOS-87-1-R

February 1987

KMOS – A FORTRAN LIBRARY FOR NONLINEAR OPTIMIZATION

J.W. Bandler, S.H. Chen and M.L. Renault

Abstract

KMOS is a library of Fortran routines for solving nonlinear optimization problems. It includes seven optimization routines, namely MMLC for linearly constrained minimax problems using exact gradients, MMAG for linearly constrained minimax problems using approximate gradients, L1LC for linearly constrained ℓ_1 problems using exact gradients, L1AG for linearly constrained ℓ_1 problems using approximate gradients, S1LC for linearly constrained one-sided ℓ_1 problems using exact gradients, S1AG for linearly constrained one-sided ℓ_1 problems using approximate gradients and L2OS for unconstrained least-squares problems using exact gradients. The general theory behind these algorithms has been described by Madsen. The basic iteration uses either a first-order method to solve a linearized subproblem or a quasi-Newton method to solve the appropriate optimality equations. The KMOS library is developed to provide a unified interface to a user's program and a standardized printing service. It has also significantly reduced the size of the combined Fortran codes because the optimizers share many common subroutines.

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant A7239.

The authors are with the Simulation Optimization Systems Research Laboratory and the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada L8S 4L7.

I. INTRODUCTION

KMOS is a library of Fortran routines for solving nonlinear optimization problems. It includes seven optimization routines, namely MMLC for linearly constrained minimax problems using exact gradients [1],[2],[3], MMAG for linearly constrained minimax problems using approximate gradients [4],[5],[6], L1LC for linearly constrained ℓ_1 problems using exact gradients [7],[8],[9], L1AG for linearly constrained ℓ_1 problems using approximate gradients [4],[5],[6], S1LC for linearly constrained one-sided ℓ_1 problems using exact gradients, S1AG for linearly constrained one-sided ℓ_1 problems using approximate gradients and L2OS for unconstrained least-squares problems using exact gradients [10]. The general theory behind these algorithms has been described by Madsen [11]. The basic iteration uses either a first-order method to solve a linearized subproblem or a quasi-Newton method to solve the appropriate optimality equations. The KMOS library is developed to provide a unified interface to a user's program and a standardized printing service. It has also significantly reduced the size of the combined Fortran codes because the optimizers share many common subroutines.

The Fortran package MMLC for solving linearly constrained minimax problem was first developed by Bandler and Zuberek [1],[2] in 1982 for the CDC 170/730 system. Since then, many changes have taken place, both in the hardware and the software. For hardware, VAX systems have replaced the original CDC machine. The programming language has been upgraded to the current Fortran 77. Most importantly, several new optimization algorithms have been developed and implemented. These include the linearly constrained ℓ_1 package [7],[8],[9], the minimax and ℓ_1 packages using approximate gradients [4],[5],[6], the 2-stage least-squares package [10] and the one-sided ℓ_1 packages. It was felt that in order to facilitate applications in the future the services of these optimizers should be made available in a standardized format. The KMOS library was thus created.

The standard calling sequence to the optimization routines follows exactly that of the original MMLC package. A uniform printing format for reporting intermediate and final results of optimization is provided. This will undoubtedly make it much easier to apply different optimization methods at the same time. The user only needs to make minimal changes to his/her program and therefore is much less likely to get confused. Also, the size of compact KMOS is much smaller than the combined size of the separate packages since they share many common subroutines.

KMOS is written in Fortran 77 for the VAX machine with VMS operating system. In order to utilize the library, the user should

1) write a Fortran program which prepares the relevant parameters and sets up a proper call to an optimization routine in KMOS (see the following sections);

2) compile this program using

$FORTRAN user__program

3) link the object code with the KMOS library using

$LINK user__program+KMOS/LIB

Notice that the name "user__program" is only symbolic. Certainly the user may instead use other names or the name may represent several Fortran modules edited and compiled separately.

II. GENERAL DESCRIPTION

Given a set of nonlinear functions

$$f_j(\mathbf{x}), \ j = 1, ..., m$$

of n variables

$$\mathbf{x} = [x_1 \ x_2 \ .... \ x_n]^T,$$

we try to find a local minimum of the objective function $F(\mathbf{x})$ which is defined in the minimax, least-squares, ℓ_1 or one-sided ℓ_1 sense. Except for the case of least-squares, the present packages can also minimize $F(\mathbf{x})$ subject to linear constraints

$$\mathbf{c}_k^T \mathbf{x} + b_k = 0, \qquad k = 1, ..., \ell_{eq},$$

$$\mathbf{c}_k^T \mathbf{x} + b_k \geq 0, \qquad k = \ell_{eq} + 1, ..., \ell,$$

where c_k and b_k are constants.

The minimax objective function is defined as

$$F(\mathbf{x}) = \max_j \{f_j(\mathbf{x})\}.$$

The least-squares objective function is defined as

$$F(\mathbf{x}) = \sum_{j=1}^{m} [f_j(\mathbf{x})]^2 .$$

The ℓ_1 objective function is defined as

$$F(\mathbf{x}) = \sum_{j=1}^{m} |f_j(\mathbf{x})| .$$

The one-sided ℓ_1 objective function is defined as

$$F(\mathbf{x}) = \sum_{j \varepsilon J} f_j(\mathbf{x}) .$$

where $J = \{j \,|\, f_j(\mathbf{x}) \geq 0\}$.

III. LIST OF ARGUMENTS

It is utterly important for a user to declare double precision for all real values. The user is, therefore, advised to declare, in all his or her program segments,

```
IMPLICIT  REAL*8  (A-H,O-Z)
```

The subroutine call to the optimizers from a user's program is

```
CALL  MMLC1A (FDF,N,M,L,LEQ,B,C,LC,X,DX,EPS,MAXF,KEQS,W,IW,ICH,IPR,IFALL)
CALL  MMAG1A (FUN,N,M,L,LEQ,B,C,LC,X,DX,EPS,MAXF,KEQS,W,IW,ICH,IPR,IFALL)
CALL  L1LC1A (FDF,N,M,L,LEQ,B,C,LC,X,DX,EPS,MAXF,KEQS,W,IW,ICH,IPR,IFALL)
CALL  L1AG1A (FUN,N,M,L,LEQ,B,C,LC,X,DX,EPS,MAXF,KEQS,W,IW,ICH,IPR,IFALL)
CALL  S1LC1A (FDF,N,M,L,LEQ,B,C,LC,X,DX,EPS,MAXF,KEQS,W,IW,ICH,IPR,IFALL)
CALL  S1AG1A (FUN,N,M,L,LEQ,B,C,LC,X,DX,EPS,MAXF,KEQS,W,IW,ICH,IPR,IFALL)
CALL  L2OS1A (FDF,N,M,X,DX,EPS,MAXF,KEQS,W,IW,ICH,IPR,IFALL)
```

For MMAG, L1AG and S1AG, a common block must be defined as

```
COMMON /APPROX/ IP0,IP1,IP2,IWG,IWRK(5)
```

The arguments are explained as follows.

FDF is the name of a subroutine supplied by the user for MMLC, L1LC, S1LC and L2OS. It must assume the form

```
SUBROUTINE FDF (N,M,X,DF,F)
REAL*8 X(N), DF(M,N), F(M)
```

When an optimization routine calls FDF, the variables are given in X(1), X(2), ..., X(N). FDF must calculate the values of the functions as well as their derivatives and store the results in

$$F(J) = f_J(\mathbf{x}), \qquad J = 1, \ldots, M$$

$$DF(J,I) = df_J/dx_I, \qquad I = 1, \ldots, N, \quad J = 1, \ldots, M.$$

Notice that FDF is only a symbolic name. The actual name of this subroutine is arbitrary and it must be defined in the calling program as EXTERNAL.

FUN is the name of a subroutine supplied by the user for MMAG, L1AG and S1AG. It must assume the form

```
SUBROUTINE FUN (N,M,X,F)
REAL*8 X(N), F(M)
```

When an optimization routine calls FUN, the variables are given in X(1), X(2), ..., X(N). FUN must calculate the values of the functions and store the results in

$$F(J) = f_J(\mathbf{x}), \quad J = 1, \ldots, M.$$

Notice that FUN is only a symbolic name. The actual name of this subroutine is arbitrary and it must be defined in the calling program as EXTERNAL.

N is an integer argument which must be set to the number of optimization parameters. Its value must be positive and it is not changed by the package.

M is an integer argument which must be set to the number of residual functions defining the appropriate norm's objective function. Its value must be positive and it is not changed by the package.

L is an integer argument which must be set to the total number of linear constraints including equality and inequality constraints. Its value must be positive or zero and it is not changed by the package.

LEQ is an integer argument which must be set to the number of equality constraints. LEQ must not be greater than N (otherwise the system is already over-determined), and not greater than L.

B is a real array of dimension B(LC), where argument LC is defined below. The elements B(K), K=1,...,L, must be set to the constant terms of the linear constraints (see definition of C below). The contents of B is not changed by the package.

C is a real matrix of dimension C(LC,N). It must contain the coefficients of the constraints. The Kth constraint is defined by

$$C(K,1)*X(1) + \ldots + C(K,N)*X(N) + B(K) = 0\,, \quad \text{if } K \leq LEQ,$$

$$C(K,1)*X(1) + \ldots + C(K,N)*X(N) + B(K) \geq 0\,, \quad \text{otherwise.}$$

LC is an integer argument which must be set to the first dimension of arrays B and C. It must be not less than L. If L=0, LC must be at least 1. Its value is not changed by the package.

X is a real array of dimension X(N). On entry, it must be set to the initial values of the variables (starting point) before calling KMOS. Upon return from KMOS, it contains the solution.

DX is a real variable which controls the step length of the iteration. On entry, a value between 0.05 - 0.2 is usually used. Upon return, DX contains the last value of the bound on the step length.

EPS is a real variable which specifies the required accuracy of the solution. A value between 1.D-4 to 1.D-6 is suggested. Upon return, EPS contains the length of the last step taken in the iteration.

MAXF is an integer argument which limits the maximum number of calls to FDF or FUN. Upon return, MAXF is set to the actual number of such calls.

KEQS is an integer which controls the use of quasi-Newton iterations (Stage 2). Normally, KEQS=3 is used for MMLC, L1LC, S1LC or L2OS and KEQS=5 is used for MMAG, L1AG or S1AG. Setting KEQS=MAXF will in effect disable Stage 2. Upon return, KEQS contains the number of switches to Stage 2 that have taken place.

W is a real array providing working space for KMOS routines. Its dimension is given by IW. Upon return, the first M elements of W contain the residual function values at the solution, i.e.,

$$W(I)=f_I(\mathbf{x}), \quad I=1,\ldots,M.$$

IW is an integer indicating the size of working space. The minimum size of the working array is

For MMLC: 2*M*N+5*N*N+4*M+8*N+4*LC+3

For MMAG: 3*M*N+6*N*N+5*M+10*N+4*LC+3

For L1LC and S1LC: 2*M*N+5*N*N+5*M+10*N+4*LC

For L1AG and S1AG: 3*M*N+6*N*N+6*M+12*N+4*LC

For L2OS: 3*M*N+2*N*N+4*M+9*N

It is probably advisable to use IW=3*M*N+6*N*N+6*M+12*N+4*LC which satisfies the requirement of all packages.

ICH is an integer. It must be set to the unit number for printed output generated by KMOS. The user can make a "quiet call" to KMOS by setting ICH < 0, in which case no printed message will be generated. This in effect emulates the original entries MMLC1Q, etc. Its value is not changed by the package.

IPR is an integer that controls the printed output. Suppose that |IPR| = *****#, where * or # indicates a digit, then

***** specifies the frequency of reporting the values of functions and variables in the printed output

if # $>$ 0 then partial derivatives will also be reported if IPR <0 then partial derivative verification will be performed at the starting point (ignored by MMAG, L1AG and S1AG)

Examples:

IPR = 100: to report values of the functions and variables for every 10 iterations.

IPR = -50: to verify partial derivatives and to report values of the functions and variables for every 5 iterations.

IPR = 151: to report values of the functions, variables and derivatives for every 15 iterations.

IFALL is an integer which, on return, contains information about the type of the solution.

IFALL = -2: feasible region is empty (conflicting constraints);

IFALL = -1: incorrect data (N<0, EPS<0, IW too small, etc.);

IFALL = 0: regular solution reached with required accuracy;

IFALL = 1: singular solution reached with required accuracy;

IFALL = 2: solution reached with machine accuracy;

IFALL = 3: number of calls to FDF or FUN reached MAXF;

IFALL = 4: iteration terminated by the user (see below).

MARK The user may terminate the optimization and force a return from KMOS by setting MARK=0 in FDF or FUN, where integer MARK must have been declared as

COMMON /MML000/ MARK

Arguments relating to gradient approximation are discussed as follows. They are applicable to MMAG, L1AG and S1AG and must be declared as

COMMON /APPROX/ IP0,IP1,IP2,IWG,IWRK(5)

IP0 is an integer that indicates whether the initial approximate gradient should be computed by KMOS (by setting IP0=1) or to be supplied by the user (by setting IP0=0). If IP0=0, the user must supply the approximate derivatives at the starting point in the working array W, from W(2M+1) to W(2M+N*M), as follows.

W((I+1)*M+J) = dF(J)/dX(I), I=1,...,N, J=1,...,M.

Its value is not changed by the package.

IP1 is an integer that controls the frequency of perturbations in Stage 1 (see the following Table). Its value is not changed by the package.

IP2 is an integer that controls the frequency of perturbations in Stage 2 (see Table I). Its value is not changed by the package.

TABLE I. COMBINED EFFECT OF IP1 AND IP2

	Perturbations Performed	
Arguments	Stage 1	Stage 2
IP1>IP2>0	every (IP1)th iteration	every (IP2)th iteration
IP2>IP1>0	every (IP1)th iteration	every (IP1)th iteration
IP1>0,IP2<0	every (IP1)th iteration	none
IP1<0,IP2>0	none	every (IP2)th iteration
IP1<0,IP2<0	none	none

Note: if IP2>0, perturbations are performed on entry to Stage 2

IWG is an integer that indicates whether the weighted Broyden update should be used. If IWG=0 the original Broyden formula is used. If IWG=1, the user must supply the weights in the working array W in the following order

$$W(K+(I-1)*M+J) = Weight(I,J), \quad I=1,...,N, \quad J=1,...,M,$$

where

$$K = 2*M*N+6*N*N+5*M+10*N+4*LC+3 \qquad \text{for MMAG,}$$

$$K = 2*M*N+6*N*N+6*M+12*N+4*LC \qquad \text{for L1AG and S1AG.}$$

Its value is not changed by the package.

IWRK is an integer array of dimension IWRK(5). It is used by KMOS as additional working space relating to gradient approximation.

IV. EXAMPLE

The HALD example [1] is used to illustrate the use of the optimization algorithms available in the KMOS library. The example is used in its original form for the optimization algorithms which are suited to designs, namely entries MMLC, MMAG, S1LC and S1AG. The example has been slightly converted in form for the optimization algorithms which are suited to parameter identification, namely entries L1LC, L1AG and L2OS.

<u>The HALD example in its original form used for design purposes</u>

Minimize

$$F(\mathbf{x}) = \| f_i(\mathbf{x}) \| \quad \text{for } i = 1, 2, 3$$

subject to

$$-3x_1 - x_2 - 2.5 \geq 0,$$

where

$$f_1(\mathbf{x}) = x_1^2 + x_2^2 + x_1 x_2 - 1,$$

$$f_2(\mathbf{x}) = \sin(x_1),$$

$$f_3(\mathbf{x}) = -\cos(x_2).$$

<u>The converted HALD example used for parameter identification purposes</u>

Minimize

$$F(\mathbf{x}) = \| f_i(\mathbf{x}) - \text{spec}_i \| \quad \text{for } i = 1, 2, 3$$

subject to (except for L2OS algorithm)

$$-3x_1 - x_2 - 2.5 \geq 0,$$

where

$$f_1(\mathbf{x}) = x_1^2 + x_2^2 + x_1 x_2 - 1,$$

$$f_2(\mathbf{x}) = \sin(x_1),$$

$$f_3(\mathbf{x}) = -\cos(x_2),$$

and

$$spec_i = f_i(\mathbf{x}^*) \quad \text{for } i = 1, 2, 3$$

for $\mathbf{x}^*$ denoting the minimax design solution, known apriori to be

$$\mathbf{x}^{*T} = [-.892857 \quad .178571]$$

Source code and results

The user written Fortran source code required to solve the HALD problem is listed in the following pages. The user's program is composed of the main segment which prepares parameters and calls the desired optimization algorithm of KMOS (pp. 12-14), and the segment which calculates the values of residual functions and, if required, their first partial derivatives (pp. 15-16). These user written routines are then compiled and linked to the KMOS library.

Results obtained by the optimizers are also presented:

- the original HALD example solved by the minimax algorithm using exact gradients (pp. 17-20).
- the original HALD example solved by the minimax algorithm using gradient approximations (pp. 21-23).
- the original HALD example solved by the one-sided ℓ_1 algorithm using exact gradients (pp. 24-26).
- the original HALD example solved by the one-sided ℓ_1 algorithm using gradient approximations (pp. 27-29).
- the converted HALD example solved by the ℓ_1 algorithm using exact gradients (pp. 30-32).
- the converted HALD example solved by the ℓ_1 algorithm using gradient approximations (pp. 33-35).
- the converted HALD example solved by the ℓ_2 algorithm using exact gradients (pp. 36-38).

```
      PROGRAM HALD

      implicit real*8 (a-h,o-z)

      parameter (n=2,m=3,lc=1,l=1,leq=0)
      parameter (iw=3*m*n+6*n*n+6*m+12*n+4*lc)
      dimension x(n),w(iw),b(lc),c(lc,n)
      dimension solmm(n),dum(m)
      external  fdf,fun
      common /approx/ ip0,ip1,ip2,iwg,iwrk(5)
      common /blk1/   iopt,spec(3)

c display program information

      write(*,100)
100   format(1h1,
     +/' cccccccccccccccccccccccccccccccccccccccccccccccccccccccccc',
     +/' c                                                        c',
     +/' c     HALD example for design purposes:                  c',
     +/' c                                                        c',
     +/' c  Minimize       F(x) = ‖ fi(x) ‖ norm                  c',
     +/' c                 for i=1,2,3                            c',
     +/' c                                                        c',
     +/' c  subject to     - 3*x1 - x2 - 2.5 >= 0                 c',
     +/' c                                                        c',
     +/' c  where          f1(x) = x1**2 + x2**2 + x1*x2 - 1      c',
     +/' c                 f2(x) = sin(x1)                        c',
     +/' c                 f3(x) = -cos(x2)                       c',
     +/' c                                                        c',
     +/' c                                                        c',
     +/' c                                                        c',
     +/' c                                                        c',
     +/' c     HALD example converted in form for parameter       c',
     +/' c                  identification purposes:              c',
     +/' c                                                        c',
     +/' c  Minimize       F(x) = ‖ fi(x) - speci ‖ norm          c',
     +/' c                 for i=1,2,3                            c',
     +/' c                                                        c',
     +/' c  subject to     - 3*x1 - x2 - 2.5 >= 0                 c',
     +/' c                 (except for the L2OS optimizer)        c',
     +/' c                                                        c',
     +/' c  where          f1(x) = x1**2 + x2**2 + x1*x2 - 1      c',
     +/' c                 f2(x) = sin(x1)                        c',
     +/' c                 f3(x) = -cos(x2)                       c',
     +/' c                                                        c',
     +/' c  and where      spec1 = f1(x) @ minimax design solution c',
     +/' c                 spec2 = f2(x) @ minimax design solution c',
     +/' c                 spec3 = f3(x) @ minimax design solution c',
     +/' c                                                        c',
     +/' cccccccccccccccccccccccccccccccccccccccccccccccccccccccccc')

c set solmm() to the known minimax solution and obtain the
```

```
c specifications to be used for parameter identification purposes

   iopt=0
   solmm(1) = -.8928571d0
   solmm(2) =  .1785714d0
   call fun(n,m,solmm,dum)
   do 10 i=1,m
10       spec(i)=dum(i)

c display the menu and prompt the user

20    write(*,200)
200   format(1h1,/' Available optimizers are:'/,
     +10x,'1.  MMLC  (for design purpose)'/,
     +10x,'2.  MMAG  (for design purpose)'/,
     +10x,'3.  S1LC  (for design purpose)'/,
     +10x,'4.  S1AG  (for design purpose)'/,
     +10x,'5.  L1LC  (for identification purpose)'/,
     +10x,'6.  L1AG  (for identification purpose)'/,
     +10x,'7.  L2OS  (for identification purpose)'/,
     +' Enter your choice : ',$)
   read(*,*) iopt
   if(iopt.lt.1.or.iopt.gt.7) stop

c linear inequality constraint:  - 3*x1 - x2 - 2.5 >= 0

   b(1) = -2.5D0
   c(1,1) = -3.D0
   c(1,2) = -1.D0

c initial guess

   x(1) = -2.D0
   x(2) = -1.D0

c optimization parameters

   dx = 0.1D0
   eps = 1.D-6
   maxf = 500
   keqs = 3
   ich = 6
   ipr = -500
   ip0=1
   ip1=5
   ip2=5
   iwg=0

c optimize

   if(iopt.eq.1) call mmlc1a(fdf,n,m,l,leq,b,c,lc,x,dx,eps,
     +                          maxf,keqs,w,iw,ich,ipr,ifall)
   if(iopt.eq.2) call mmag1a(fun,n,m,l,leq,b,c,lc,x,dx,eps,
     +                          maxf,keqs,w,iw,ich,ipr,ifall)
```

```
      if(iopt.eq.3) call s1lc1a(fdf,n,m,l,leq,b,c,lc,x,dx,eps,
     +                          maxf,keqs,w,iw,ich,ipr,ifall)
      if(iopt.eq.4) call s1ag1a(fun,n,m,l,leq,b,c,lc,x,dx,eps,
     +                          maxf,keqs,w,iw,ich,ipr,ifall)
      if(iopt.eq.5) call l1lc1a(fdf,n,m,l,leq,b,c,lc,x,dx,eps,
     +                          maxf,keqs,w,iw,ich,ipr,ifall)
      if(iopt.eq.6) call l1ag1a(fun,n,m,l,leq,b,c,lc,x,dx,eps,
     +                          maxf,keqs,w,iw,ich,ipr,ifall)
      if(iopt.eq.7) call l2os1a(fdf,n,m,x,dx,eps,
     +                          maxf,keqs,w,iw,ich,ipr,ifall)

      goto 20
      end
```

```
      SUBROUTINE FDF(N,M,X,DF,F)

      implicit real*8 (a-h,o-z)

      dimension x(n),f(m),df(m,n)
      common /blk1/ iopt,spec(3)

      x1 = x(1)
      x2 = x(2)

      f(1) = x1**2 + x2**2 + x1*x2 - 1.D0
      f(2) = sin(x1)
      f(3) = -cos(x2)

c if parameter identification is desired construct the error
c functions of the converted HALD problem

      if(iopt.gt.4) then
         do 10 i=1,m
10          f(i)=f(i)-spec(i)
      endif

      df(1,1) = 2.D0*x1 + x2
      df(1,2) = 2.D0*x2 + x1
      df(2,1) = cos(x1)
      df(2,2) = 0.D0
      df(3,1) = 0.D0
      df(3,2) = sin(x2)

      return
      end
```

```
      SUBROUTINE FUN(N,M,X,F)

      implicit real*8 (a-h,o-z)

      dimension x(n),f(m)
      common /blk1/ iopt,spec(3)

      x1 = x(1)
      x2 = x(2)

      f(1) = x1**2 + x2**2 + x1*x2 - 1.D0
      f(2) = sin(x1)
      f(3) = -cos(x2)

c if parameter identification is desired construct the error
c functions of the converted HALD problem

      if(iopt.gt.4) then
         do 10 i=1,m
10          f(i)=f(i)-spec(i)
      endif

      return
      end
```

```
$RUN HALD

 cccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
 c                                                        c
 c     HALD example for design purposes:                  c
 c                                                        c
 c  Minimize        F(x) = || fi(x) ||norm                c
 c                  for i=1,2,3                           c
 c                                                        c
 c  subject to      - 3*x1 - x2 - 2.5 >= 0                c
 c                                                        c
 c  where           f1(x) = x1**2 + x2**2 + x1*x2 - 1     c
 c                  f2(x) = sin(x1)                       c
 c                  f3(x) = -cos(x2)                      c
 c                                                        c
 c                                                        c
 c                                                        c
 c                                                        c
 c     HALD example converted in form for parameter       c
 c                  identification purposes:              c
 c                                                        c
 c  Minimize        F(x) = || fi(x) - speci ||norm        c
 c                  for i=1,2,3                           c
 c                                                        c
 c  subject to      - 3*x1 - x2 - 2.5 >= 0                c
 c                  (except for the L2OS optimizer)       c
 c                                                        c
 c  where           f1(x) = x1**2 + x2**2 + x1*x2 - 1     c
 c                  f2(x) = sin(x1)                       c
 c                  f3(x) = -cos(x2)                      c
 c                                                        c
 c  and where       spec1 = f1(x) @ minimax design solution  c
 c                  spec2 = f2(x) @ minimax design solution  c
 c                  spec3 = f3(x) @ minimax design solution  c
 c                                                        c
 cccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
 Available optimizers are:
          1.  MMLC  (for design purpose)
          2.  MMAG  (for design purpose)
          3.  S1LC  (for design purpose)
          4.  S1AG  (for design purpose)
          5.  L1LC  (for identification purpose)
          6.  L1AG  (for identification purpose)
          7.  L2OS  (for identification purpose)
$Enter your choice :
  Input : 1
```

```
PAGE :  1    4-FEB-1987    15:06:57      MMLC8D PACKAGE V:87.01
LINEARLY CONSTRAINED MINIMAX OPTIMIZATION

 INPUT DATA
 ----------

NUMBER OF VARIABLES (N) . . . . . . . . . . . . . . . . . . . . . . . . .2

NUMBER OF FUNCTIONS (M) . . . . . . . . . . . . . . . . . . . . . . . . .3

TOTAL NUMBER OF LINEAR CONSTRAINTS (L)  . . . . . . . . . . . . . . . . .1

NUMBER OF EQUALITY CONSTRAINTS (LEQ)  . . . . . . . . . . . . . . . . . .0

STEP LENGTH (DX)  . . . . . . . . . . . . . . . . . . . . . . . .1.000E-01

ACCURACY (EPS)  . . . . . . . . . . . . . . . . . . . . . . . .  1.000E-06

MAX NUMBER OF FUNCTION EVALUATIONS (MAXF)  . . . . . . . . . . . .  500

NUMBER OF SUCCESSIVE ITERATIONS (KEQS)  . . . . . . . . . . . . .   . .3

WORKING SPACE (IW) . . . . . . . . . . . . . . . . . . . . . . .   . .88

PRINTOUT CONTROL (IPR) . . . . . . . . . . . . . . . . . . . . . . .-500


VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.

FUNCTION EVALUATION :    1 / 0

                           MINIMAX OBJECTIVE:   6.000000000000E+00

                         VARIABLES                  FUNCTION VALUES

                 1  -2.000000000000E+00        1    6.000000000000E+00
                 2  -1.000000000000E+00        2  -9.092974268257E-01
                                               3  -5.403023058681E-01


SOLUTION
--------
                            MINIMAX OBJECTIVE:  -3.303571428571E-01

                           VARIABLES                FUNCTION VALUES

                 1  -8.928571428571E-01      1  -3.303571428571E-01
                 2    1.785714285714E-01     2  -7.788668934368E-01
                                             3  -9.840984453126E-01
```

PAGE : 2   4-FEB-1987   15:06:57   MMLC8D PACKAGE V:87.01
LINEARLY CONSTRAINED MINIMAX OPTIMIZATION

TYPE OF SOLUTION (IFALL) 2

NUMBER OF FUNCTION EVALUATIONS 10

NUMBER OF SHIFTS TO STAGE-2 2

EXECUTION TIME (IN SECONDS) 0.040

```
Available optimizers are:
         1.  MMLC  (for design purpose)
         2.  MMAG  (for design purpose)
         3.  S1LC  (for design purpose)
         4.  S1AG  (for design purpose)
         5.  L1LC  (for identification purpose)
         6.  L1AG  (for identification purpose)
         7.  L2OS  (for identification purpose)
$Enter your choice :
  Input : 2
```

```
PAGE :  1    4-FEB-1987   15:07:08     MMAG8D PACKAGE V:87.01
LINEARLY CONSTRAINED MINIMAX OPTIMIZATION WITH GRADIENT APPROXIMATION


INPUT DATA
----------

NUMBER OF VARIABLES (N) . . . . . . . . . . . . . . . . . . . . . . . . . .2

NUMBER OF FUNCTIONS (M) . . . . . . . . . . . . . . . . . . . . . . . . . .3

TOTAL NUMBER OF LINEAR CONSTRAINTS (L)  . . . . . . . . . . . . . . . . . .1

NUMBER OF EQUALITY CONSTRAINTS (LEQ)  . . . . . . . . . . . . . . . . . . .0

STEP LENGTH (DX)  . . . . . . . . . . . . . . . . . . . . . . . . .1.000E-01

ACCURACY (EPS)  . . . . . . . . . . . . . . . . . . . . . . . . . .1.000E-06

MAX NUMBER OF FUNCTION EVALUATIONS (MAXF)  . . . . . . . . . . . . . . . 500

INITIAL GRADIENT APPROXIMATION FLAG  (IPO) . . . . . . . . . . . . . . . . 1

FREQUENCY OF PERTURBATIONS IN STAGE1 (IP1). . . . . . . . . . . . . . . . .5

FREQUENCY OF PERTURBATIONS IN STAGE2 (IP2). . . . . . . . . . . . . . . . .5

WEIGHTED OR NON-WEIGHTED FORMULA (IWEIGH) . . . . . . . . . . . . . . . . .0

NUMBER OF SUCCESSIVE ITERATIONS (KEQS)  . . . . . . . . . . . . . . . . . .3

WORKING SPACE (IW) . . . . . . . . . . . . . . . . . . . . . . . . . . . .88

PRINTOUT CONTROL (IPR) . . . . . . . . . . . . . . . . . . . . . . . . .-500


FUNCTION EVALUATION :    1 / 0
                              MINIMAX OBJECTIVE:  6.000000000000E+00

                             VARIABLES                FUNCTION VALUES

                      1  -2.000000000000E+00      1    6.000000000000E+00
                      2  -1.000000000000E+00      2  -9.092974268257E-01
                                                  3  -5.403023058681E-01

 SOLUTION
 --------

                              MINIMAX OBJECTIVE:  -3.303571428570E-01

                           VARIABLES                 FUNCTION VALUES

                     1  -8.928569798009E-01      1  3.303571148570E-01
                     2    1.785709394026E-01     2  7.788667911696E-01
                                                 3  9.840985322006E-01
```

```
PAGE :  2    4-FEB-1987   15:07:08     MMAG8D PACKAGE V:87.01
LINEARLY CONSTRAINED MINIMAX OPTIMIZATION WITH GRADIENT APPROXIMATION


TYPE OF SOLUTION (IFALL) . . . . . . . . . . . . . . . . . . . . . . . . 1

NUMBER OF ORDINARY ITERATIONS  . . . . . . . . . . . . . . . . . . . . .41

NUMBER OF SPECIAL ITERATIONS . . . . . . . . . . . . . . . . . . . . . . 7

NUMBER OF PERTURBATIONS  . . . . . . . . . . . . . . . . . . . . . . . .10

TOTAL NUMBER OF FUNCTION EVALUATIONS . . . . . . . . . . . . . . . . . .68

NUMBER OF SHIFTS TO STAGE-2 . . . . . . . . . . . . . . . . . . . . . . 4

EXECUTION TIME (IN SECONDS) . . . . . . . . . . . . . . . . . . . . . 0.120
```

```
 Available optimizers are:
          1.  MMLC  (for design purpose)
          2.  MMAG  (for design purpose)
          3.  S1LC  (for design purpose)
          4.  S1AG  (for design purpose)
          5.  L1LC  (for identification purpose)
          6.  L1AG  (for identification purpose)
          7.  L2OS  (for identification purpose)
$Enter your choice :
  Input : 3
```

```
PAGE :  1    4-FEB-1987   15:07:22     S1LC8D PACKAGE V:87.01
LINEARLY CONSTRAINED ONE-SIDED L1 OPTIMIZATION

INPUT DATA
----------

NUMBER OF VARIABLES (N) . . . . . . . . . . . . . . . . . . . . . . . .     2

NUMBER OF FUNCTIONS (M) . . . . . . . . . . . . . . . . . . . . . . . .     3

TOTAL NUMBER OF LINEAR CONSTRAINTS (L)  . . . . . . . . . . . . . . . .     1

NUMBER OF EQUALITY CONSTRAINTS (LEQ)  . . . . . . . . . . . . . . . . .     0

STEP LENGTH (DX)  . . . . . . . . . . . . . . . . . . . . . . . 1.000E-01

ACCURACY (EPS)  . . . . . . . . . . . . . . . . . . . . . . . . 1.000E-06

MAX NUMBER OF FUNCTION EVALUATIONS (MAXF)  . . . . . . . . . . . . . . . 500

NUMBER OF SUCCESSIVE ITERATIONS (KEQS)  . . . . . . . . . . . . . . . .     3

WORKING SPACE (IW) . . . . . . . . . . . . . . . . . . . . . . . . . .    88

PRINTOUT CONTROL (IPR) . . . . . . . . . . . . . . . . . . . . . . .  -500

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.

FUNCTION EVALUATION :    1 / 0
                             ONE-SIDED L1 OBJECTIVE:    6.000000000000E+00

                            VARIABLES                  FUNCTION VALUES

                      1  -2.000000000000E+00       1    6.000000000000E+00
                      2  -1.000000000000E+00       2   -9.092974268257E-01
                                                   3   -5.403023058681E-01


SOLUTION
--------
                            ONE-SIDED L1 OBJECTIVE:    2.775557561563E-17

                            VARIABLES                  FUNCTION VALUES

                      1  -9.575049304629E-01       1    2.775557561563E-17
                      2  -8.016455106386E-02       2   -8.177580476547E-01
                                                   3   -9.967885427595E-01
```

```
PAGE :  2    4-FEB-1987   15:07:22     S1LC8D PACKAGE V:87.01
LINEARLY CONSTRAINED ONE-SIDED L1 OPTIMIZATION


TYPE OF SOLUTION (IFALL)  . . . . . . . . . . . . . . . . . . . . . . . .2

NUMBER OF FUNCTION EVALUATIONS . . . . . . . . . . . . . . . . . . . . . 9

NUMBER OF SHIFTS TO STAGE-2 . . . . . . . . . . . . . . . . . . . . . . .0

EXECUTION TIME (IN SECONDS) . . . . . . . . . . . . . . . . . . . . .0.050
```

```
 Available optimizers are:
         1.  MMLC  (for design purpose)
         2.  MMAG  (for design purpose)
         3.  S1LC  (for design purpose)
         4.  S1AG  (for design purpose)
         5.  L1LC  (for identification purpose)
         6.  L1AG  (for identification purpose)
         7.  L2OS  (for identification purpose)
$Enter your choice :
  Input : 4
```

```
PAGE :  1    4-FEB-1987   15:07:33     S1AG8D PACKAGE V:87.01
LINEARLY CONSTRAINED ONE-SIDED L1 OPTIMIZATION WITH GRADIENT APPROXIMATION

INPUT DATA
----------

NUMBER OF VARIABLES (N) . . . . . . . . . . . . . . . . . . . . . . . .    2

NUMBER OF FUNCTIONS (M) . . . . . . . . . . . . . . . . . . . . . . . .    3

TOTAL NUMBER OF LINEAR CONSTRAINTS (L)  . . . . . . . . . . . . . . . .    1

NUMBER OF EQUALITY CONSTRAINTS (LEQ)  . . . . . . . . . . . . . . . . .    0

STEP LENGTH (DX)  . . . . . . . . . . . . . . . . . . . . . . . . 1.000E-01

ACCURACY (EPS)  . . . . . . . . . . . . . . . . . . . . . . . . . 1.000E-06

MAX NUMBER OF FUNCTION EVALUATIONS (MAXF)  . . . . . . . . . . . . . . . 500

INITIAL GRADIENT APPROXIMATION FLAG  (IPO) . . . . . . . . . . . . . . .   1

FREQUENCY OF PERTURBATIONS IN STAGE1 (IP1). . . . . . . . . . . . . . .    5

FREQUENCY OF PERTURBATIONS IN STAGE2 (IP2). . . . . . . . . . . . . . .    5

WEIGHTED OR NON-WEIGHTED FORMULA (IWEIGH) . . . . . . . . . . . . . . .    0

NUMBER OF SUCCESSIVE ITERATIONS (KEQS)  . . . . . . . . . . . . . . . .    3

WORKING SPACE (IW) . . . . . . . . . . . . . . . . . . . . . . . . . .    88

PRINTOUT CONTROL (IPR) . . . . . . . . . . . . . . . . . . . . . . . .  -500

FUNCTION EVALUATION :    1 / 0
                        ONE-SIDED L1 OBJECTIVE:   6.000000000000E+00

                          VARIABLES                   FUNCTION VALUES

                   1  -2.000000000000E+00       1   6.000000000000E+00
                   2  -1.000000000000E+00       2  -9.092974268257E-01
                                                3  -5.403023058681E-01


SOLUTION
--------
                        ONE-SIDED L1 OBJECTIVE:   1.743050148661E-14

                          VARIABLES                   FUNCTION VALUES

                   1  -9.621201098209E-01       1   1.743050148661E-14
                   2  -7.188087385925E-02       2  -8.204056516677E-01
                                                3  -9.974176821468E-01
```

PAGE : 2 4-FEB-1987 15:07:33 S1AG8D PACKAGE V:87.01
LINEARLY CONSTRAINED ONE-SIDED L1 OPTIMIZATION WITH GRADIENT APPROXIMATION

TYPE OF SOLUTION (IFALL) 2

NUMBER OF ORDINARY ITERATIONS 10

NUMBER OF SPECIAL ITERATIONS 2

NUMBER OF PERTURBATIONS 2

TOTAL NUMBER OF FUNCTION EVALUATIONS 14

NUMBER OF SHIFTS TO STAGE-2 0

EXECUTION TIME (IN SECONDS) 0.060

```
 Available optimizers are:
         1.  MMLC  (for design purpose)
         2.  MMAG  (for design purpose)
         3.  S1LC  (for design purpose)
         4.  S1AG  (for design purpose)
         5.  L1LC  (for identification purpose)
         6.  L1AG  (for identification purpose)
         7.  L2OS  (for identification purpose)
$Enter your choice :
  Input : 5
```

```
PAGE :  1    4-FEB-1987   15:07:47      L1LC8D PACKAGE V:87.01
LINEARLY CONSTRAINED L1 OPTIMIZATION

INPUT DATA
----------

NUMBER OF VARIABLES (N) . . . . . . . . . . . . . . . . . . . . . .      2

NUMBER OF FUNCTIONS (M) . . . . . . . . . . . . . . . . . . . . . .      3

TOTAL NUMBER OF LINEAR CONSTRAINTS (L)  . . . . . . . . . . . . . .      1

NUMBER OF EQUALITY CONSTRAINTS (LEQ)  . . . . . . . . . . . . . . .      0

STEP LENGTH (DX)  . . . . . . . . . . . . . . . . . . . . . . . 1.000E-01

ACCURACY (EPS)  . . . . . . . . . . . . . . . . . . . . . . . . 1.000E-06

MAX NUMBER OF FUNCTION EVALUATIONS (MAXF)  . . . . . . . . . . . . . 500

NUMBER OF SUCCESSIVE ITERATIONS (KEQS)  . . . . . . . . . . . . . .    3

WORKING SPACE (IW) . . . . . . . . . . . . . . . . . . . . . . . .    88

PRINTOUT CONTROL (IPR) . . . . . . . . . . . . . . . . . . . . . .  -500

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.

FUNCTION EVALUATION :    1 / 0
                                      L1 OBJECTIVE:    6.904583901216E+00

                         VARIABLES                     FUNCTION VALUES

              1   -2.000000000000E+00          1    6.330357196429E+00
              2   -1.000000000000E+00          2   -1.304305602684E-01
                                               3    4.437961445195E-01


SOLUTION
--------

                                      L1 OBJECTIVE:    7.133382227964E-08

                         VARIABLES                     FUNCTION VALUES

              1   -8.928571000000E-01          1    5.357143999241E-08
              2    1.785713000000E-01          2    0.000000000000E+00
                                               3   -1.776238228723E-08
```

```
PAGE :  2    4-FEB-1987   15:07:47    L1LC8D PACKAGE V:87.01
LINEARLY CONSTRAINED L1 OPTIMIZATION


TYPE OF SOLUTION (IFALL) . . . . . . . . . . . . . . . . . . . . . . . .      0

NUMBER OF FUNCTION EVALUATIONS . . . . . . . . . . . . . . . . . . . . .     10

NUMBER OF SHIFTS TO STAGE-2 . . . . . . . . . . . . . . . . . . . . . .      0

EXECUTION TIME (IN SECONDS) . . . . . . . . . . . . . . . . . . . . . .  0.040
```

```
 Available optimizers are:
         1.  MMLC  (for design purpose)
         2.  MMAG  (for design purpose)
         3.  S1LC  (for design purpose)
         4.  S1AG  (for design purpose)
         5.  L1LC  (for identification purpose)
         6.  L1AG  (for identification purpose)
         7.  L2OS  (for identification purpose)
$Enter your choice :
  Input : 6
```

```
PAGE :  1    4-FEB-1987    15:07:58      L1AG8D PACKAGE V:87.01
LINEARLY CONSTRAINED L1 OPTIMIZATION WITH GRADIENT APPROXIMATION

INPUT DATA
----------

NUMBER OF VARIABLES (N) . . . . . . . . . . . . . . . . . . . . . . . .    2

NUMBER OF FUNCTIONS (M) . . . . . . . . . . . . . . . . . . . . . . . .    3

TOTAL NUMBER OF LINEAR CONSTRAINTS (L)  . . . . . . . . . . . . . . . .    1

NUMBER OF EQUALITY CONSTRAINTS (LEQ)  . . . . . . . . . . . . . . . . .    0

STEP LENGTH (DX)  . . . . . . . . . . . . . . . . . . . . . . . . 1.000E-01

ACCURACY (EPS)  . . . . . . . . . . . . . . . . . . . . . . . . . 1.000E-06

MAX NUMBER OF FUNCTION EVALUATIONS (MAXF)  . . . . . . . . . . . . . . . 500

INITIAL GRADIENT APPROXIMATION FLAG  (IP0) . . . . . . . . . . . . . . .   1

FREQUENCY OF PERTURBATIONS IN STAGE1 (IP1). . . . . . . . . . . . . . .    5

FREQUENCY OF PERTURBATIONS IN STAGE2 (IP2). . . . . . . . . . . . . . .    5

WEIGHTED OR NON-WEIGHTED FORMULA (IWEIGH) . . . . . . . . . . . . . . .    0

NUMBER OF SUCCESSIVE ITERATIONS (KEQS)  . . . . . . . . . . . . . . . .    3

WORKING SPACE (IW) . . . . . . . . . . . . . . . . . . . . . . . . . .    88

PRINTOUT CONTROL (IPR) . . . . . . . . . . . . . . . . . . . . . . . .  -500

FUNCTION EVALUATION :    1 / 0
                                      L1 OBJECTIVE:    6.904583901216E+00

                              VARIABLES                   FUNCTION VALUES

                       1   -2.000000000000E+00       1    6.330357196429E+00
                       2   -1.000000000000E+00       2   -1.304305602684E-01
                                                     3    4.437961445195E-01


SOLUTION
--------
                                      L1 OBJECTIVE:    7.133382229352E-08

                              VARIABLES                   FUNCTION VALUES

                       1   -8.928571000000E-01       1    5.357143999241E-08
                       2    1.785713000000E-01       2   -1.110223024625E-16
```

3 -1.776238219009E-08

PAGE : 2 4-FEB-1987 15:07:58 L1AG8D PACKAGE V:87.01
LINEARLY CONSTRAINED L1 OPTIMIZATION WITH GRADIENT APPROXIMATION

TYPE OF SOLUTION (IFALL)	0
NUMBER OF ORDINARY ITERATIONS	12
NUMBER OF SPECIAL ITERATIONS	3
NUMBER OF PERTURBATIONS	3
TOTAL NUMBER OF FUNCTION EVALUATIONS	21
NUMBER OF SHIFTS TO STAGE-2	0
EXECUTION TIME (IN SECONDS)	0.060

```
 Available optimizers are:
          1.  MMLC  (for design purpose)
          2.  MMAG  (for design purpose)
          3.  S1LC  (for design purpose)
          4.  S1AG  (for design purpose)
          5.  L1LC  (for identification purpose)
          6.  L1AG  (for identification purpose)
          7.  L2OS  (for identification purpose)
$Enter your choice :
  Input : 7
```

```
PAGE :  1    4-FEB-1987   15:34:38     L2OS8D PACKAGE V:87.01
UNCONSTRAINED LEAST-SQUARES OPTIMIZATION

INPUT DATA
----------

NUMBER OF VARIABLES (N) . . . . . . . . . . . . . . . . . . . . . . . .      2

NUMBER OF FUNCTIONS (M) . . . . . . . . . . . . . . . . . . . . . . . .      3

STEP LENGTH (DX)  . . . . . . . . . . . . . . . . . . . . . . . .  1.000E-01

ACCURACY (EPS)  . . . . . . . . . . . . . . . . . . . . . . . . .  1.000E-06

MAX NUMBER OF FUNCTION EVALUATIONS (MAXF)  . . . . . . . . . . . . . . . . 500

NUMBER OF SUCCESSIVE ITERATIONS (KEQS)  . . . . . . . . . . . . . . . .      3

WORKING SPACE (IW) . . . . . . . . . . . . . . . . . . . . . . . . . .      88

PRINTOUT CONTROL (IPR) . . . . . . . . . . . . . . . . . . . . . . . .    -500

VERIFICATION OF PARTIAL DERIVATIVES PERFORMED.

FUNCTION EVALUATION :    1 / 0
                           LEAST-SQUARES OBJECTIVE:    4.028738938332E+01

                             VARIABLES                      FUNCTION VALUES

                     1  -2.000000000000E+00           1    6.330357196429E+00
                     2  -1.000000000000E+00           2  -1.304305602684E-01
                                                      3    4.437961445195E-01

SOLUTION
--------
                           LEAST-SQUARES OBJECTIVE:    7.000233645414E-14

                             VARIABLES                      FUNCTION VALUES

                     1  -8.928568040142E-01           1    6.164087730520E-08
                     2    1.785703969812E-01          2    1.856391597388E-07
                                                      3  -1.781595943828E-07

TYPE OF SOLUTION (IFALL)  . . . . . . . . . . . . . . . . . . . . . . . .     0

NUMBER OF FUNCTION EVALUATIONS  . . . . . . . . . . . . . . . . . . . . .    11

NUMBER OF SHIFTS TO STAGE-2 . . . . . . . . . . . . . . . . . . . . . . .     0

EXECUTION TIME (IN SECONDS) . . . . . . . . . . . . . . . . . . . . . .  0.030
```

```
 Available optimizers are:
          1.  MMLC  (for design purpose)
          2.  MMAG  (for design purpose)
          3.  S1LC  (for design purpose)
          4.  S1AG  (for design purpose)
          5.  L1LC  (for identification purpose)
          6.  L1AG  (for identification purpose)
          7.  L2OS  (for identification purpose)
$Enter your choice :
  Input : 0

 FORTRAN STOP
```

REFERENCES

[1] J. Hald (Adapted and Edited by J.W. Bandler and W.M. Zuberek), "MMLA1Q – A Fortran package for linearly constrained minimax optimization", Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, Report SOS-81-14-UL, 1981.

[2] J.W. Bandler and W.M. Zuberek, "MMLC – A Fortran package for linearly constrained minimax optimization", Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, Report SOS-82-5-U2, 1983.

[3] J.W. Bandler, W. Kellermann and K. Madsen, "A superlinearly convergent minimax algorithm for microwave circuit design", IEEE Trans. Microwave Theory Tech., vol. MTT-33, pp. 1519-1530, 1985.

[4] J.W. Bandler, S.H. Chen, S. Daijavad and K. Madsen, "Efficient gradient approximations for nonlinear optimization of circuits and systems", Proc. IEEE Int. Symp. Circuits and Systems (San Jose, CA), pp. 964-967, 1986.

[5] J.W. Bandler, S.H. Chen, S. Daijavad and K. Madsen, "Efficient optimization with integrated gradient approximations, Part I: algorithms", Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, Report SOS-86-12-R, 1986.

[6] J.W. Bandler, S.H. Chen, S. Daijavad and K. Madsen, "Efficient optimization with integrated gradient approximations, Part II: implementation", Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, Report SOS-86-13-R, 1986.

[7] J. Hald, "A 2-stage algorithm for nonlinear ℓ_1 optimization", Report No. NI-81-03, Institute for Numerical Analysis, Technical University of Denmark, Lyngby, Denmark, 1981.

[8] J. Hald and K. Madsen, "Combined LP and quasi-Newton methods for nonlinear ℓ_1 optimization", SIAM J. on Numerical Analysis, vol. 22, pp.68-80, 1985.

[9] J.W. Bandler, W. Kellermann and K. Madsen, "A nonlinear ℓ_1 optimization algorithm for design, modelling and diagnosis of networks", IEEE Trans. Circuits and Systems, vol. CAS-34, pp. 174-181, 1987.

[10] K. Madsen, Lecture notes and private communications, 1986/87.

[11] K. Madsen, "Minimization of Non-linear Approximation Functions", Thesis for den tekniske doktorgrad, Institute for Numerical Analysis, Technical University of Denmark, Lyngby, Denmark, 1986.