



HAL
open science

A METHOD TO MINIMIZE EXACTLY A BROAD CLASS OF NONLINEAR FUNCTIONS

Benoît Bruet

► **To cite this version:**

Benoît Bruet. A METHOD TO MINIMIZE EXACTLY A BROAD CLASS OF NONLINEAR FUNCTIONS. École Centrale de Paris. 1994. hal-04580871

HAL Id: hal-04580871

<https://hal.science/hal-04580871>

Submitted on 21 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

A METHOD TO MINIMIZE EXACTLY A BROAD CLASS OF NONLINEAR FUNCTIONS

B. BRUET^{*}

Laboratoire Productique et Logistique
Ecole Centrale des Arts et Manufactures
92290 CHATENAY-MALABRY Cedex, FRANCE
E-mail: 100321.1407@compuserve.com

Abstract

A technique is presented that minimizes exactly (in a finite number of steps) the class of nonlinear functions whose isovalue surfaces are homothetic with respect to the minimum ("HIS" functions). The method can be applied to general nonlinear functions, only leaving out finite termination. Various numerical experiments have been performed and their results compared to those obtained with the well known BFGS and conjugate gradient methods. The method proved to be more efficient than others when applied to very large problems such as those found in structural design optimization.

Keywords : nonlinear optimization, unconstrained optimization, nonlinear scaling, finite termination, homothetic surfaces, conjugate gradient, BFGS

1. DEFINITION AND MINIMIZATION OF "HIS" FUNCTIONS

Let R^{*+} be the set of strictly positive real numbers and let $F : R^n \rightarrow R$ be a function having continuous first derivatives and a unique minimum at origin such that :

$$(1) \quad F(\mu X) = G(\mu, F(X)) \quad \forall X \in R^n, \quad \forall \mu \in R^{*+}$$

where $G : R^{*+} \times R$ has continuous first derivatives G'_1 and G'_2 such that :

$$G'_1(\mu, \alpha) > 0 \quad \forall \mu \in R^{*+}, \quad \forall \alpha \in R, \quad \alpha > F(0),$$

*Ingénieur ECP - Docteur-Ingénieur de l'Université Pierre et Marie Curie

$$G'_2(\mu, \alpha) > 0 \quad \forall \mu \in \mathbb{R}^{*+}, \quad \forall \alpha \in \mathbb{R}, \quad \alpha > F(0).$$

The mathematical properties of such functions will be reviewed in further detail in section 4, but for now, let us simply state that such a function has homothetic isovalue surfaces (HIS function) and that its gradient satisfies:

$$(2) \quad F'(X) \cdot X = G'_1(1, F(X))$$

Let be $f : \mathbb{R}^n \rightarrow \mathbb{R}$ having a minimum x^* , and defined by :

$$f(x) = F(x - x^*)$$

where $F : \mathbb{R}^n \rightarrow \mathbb{R}$ is a HIS function as defined above.

This is simply a translation from the origin of coordinates to the minimum, so the results derived in section 4 still hold when applied to the minimum instead of the origin. So, f will be said to be a HIS function with respect to its unique minimum x^* .

Definition (1) then becomes :

$$(3) \quad f(x^* + \mu(x - x^*)) = G(\mu, f(x)) \quad \forall x \in \mathbb{R}^n, \quad \forall \mu \in \mathbb{R}^{*+}$$

where $G : \mathbb{R}^{*+} \times \mathbb{R}$ has continuous first derivatives such that :

$$\begin{aligned} G'_1(\mu, \alpha) &> 0 \quad \forall \mu \in \mathbb{R}^{*+}, \quad \forall \alpha \in \mathbb{R}, \\ G'_2(\mu, \alpha) &> 0 \quad \forall \mu \in \mathbb{R}^{*+}, \quad \forall \alpha \in \mathbb{R}, \end{aligned}$$

where G'_1 and G'_2 denotes the derivatives with respect to the variables.

Similarly, equation (3) yields :

$$(4) \quad f'(x) \cdot (x - x^*) = G'_1(1, f(x))$$

In this section, we will present a method to minimize exactly such HIS functions.

1.1. Overview of the method

Assume we start at a given point x and we perform an unidimensional search along a descent direction d , i.e. we restrict our attention to points y verifying :

$$(D) \quad y = x + \alpha d$$

Since d is supposed to be a descent direction for f when starting from x , we have necessarily $\alpha \geq 0$.

Similarly to the method described in [1], let us search "uphill" (that is, for increasing values of f) for a point $y \neq x$ such that :

$$f(y) = f(x)$$

As we will show in section 4.2, such a point necessarily exists for a HIS function. Let us now derive a more usable form of equation (4). After determining a point y such that $f(y)=f(x)$, we have $G'_1(1, f(y))=G'_1(f(x))$ and according to equation (4) we can write :

$$f'(y) \cdot (y-x^*) = f'(x) \cdot (x-x^*)$$

Reordering, we get :

$$(5) \quad (f'(y)-f'(x)) \cdot x^* = f'(y) \cdot y - f'(x) \cdot x$$

where $y = x + \alpha d$ for some $\alpha \in \mathbb{R}$

This form could be used directed by constructing a set of n independent equations corresponding to $n+1$ points giving the same value to f . This linear system could then be solved using any relevant method of linear algebra such as the classic Gauss method. However, this gives a poorly performing method when applied to non HIS nonlinear functions, and we will now develop a more attractive approach.

Let us find a point z lying on (D) such that :

$$(6) \quad (f'(y)-f'(x)) \cdot z = f'(y) \cdot y - f'(x) \cdot x$$

Writing $z=x+\beta d$ and remembering that $y=x+\alpha d$, we get :

$$(f'(y)-f'(x)) \cdot (x+\beta d) = f'(y) \cdot (x+\alpha d) - f'(x) \cdot x$$

Reordering and simplifying this equation yields :

$$(7) \quad \beta = \alpha \frac{f'(y) \cdot d}{f'(y) \cdot d - f'(x) \cdot d}$$

It is interesting to note that on general HIS functions, there is no relation between the z point determined as above, and the unidirectional minimum of f from x along d : point z does not necessarily need to be this minimum, and usually is simply not.

By hypothesis, d is a descent direction from x , so that $f'(x) \cdot d$ should always be negative. Similarly, by construction y is reached "uphill" along d , so $f'(y) \cdot d$ should be always positive. Therefore, the denominator of the above expression should never vanish, even for non HIS functions. Furthermore, this also implies that the denominator is superior to the numerator, yielding :

$$0 < \frac{f'(y) \cdot d}{f'(y) \cdot d - f'(x) \cdot d} < 1$$

This means that, as one could expect, point z is located somewhere between x and y , on the line joining x to y .

Using point z as determined above and subtracting (6) from (5) yields :

$$(f'(y)-f'(x)) \cdot (x^*-z) = 0$$

This means that x^* lies in the hyperplane orthogonal to $f'(y)-f'(x)$ and passing through z .

The idea of the method is then to restrict the remaining searches to proceed within this hyperplane, and then to apply the same method starting at the point z previously computed. Therefore, the searches will be restricted to linear subspaces of a dimension decreased by one at each iteration thus leading to termination in a finite number of steps (at most n).

Up to now, there has been no imposed way to choose the search directions, provided these are kept orthogonal to the subspace generated by the gradient differences $f'(y) - f'(x)$ issued from previous steps. The trivial choice is then obviously to choose the new direction of search as the projection of the opposite of the gradient at the current iteration point onto the subspace complementary to the subspace generated by gradient differences.

This is accomplished by orthogonalizing these differences through a process like the well known Gram-Schmidt procedure. Note that if the set of previous gradient differences is already orthogonalized, orthogonalizing a new gradient difference will cost only $O(n^2)$ elementary operations instead of $O(n^3)$ operations required for the classic Gram-Schmidt procedure. The projected gradient is then obtained by subtracting from it its individual projections onto the orthogonalized gradient differences.

Our method can now be summarized as follows :

- choose an initial point x_0 and the first search direction
 $d_0 = -f'(x_0) ;$
- at iteration $k \geq 0$, make an linear search along d_k to find the point $y_k \neq x_k$ such that $f(y_k) = f(x_k) ;$
- orthogonalize the gradient difference $f'(y_k) - f'(x_k)$ to previous gradient differences (if any) and normalize it ;
- compute $z_k = x_k + \beta d_k$ where β is determined by formula (7) ;
- compute the gradient $f'(z_k) ;$
- test for termination with an appropriate criterion (n iterations done, null gradient, little changing f or x, etc) ;
- if not terminated, set $x_{k+1} = z_k$ and $d_{k+1} =$ projection of $f'(z_k)$ onto the subspace orthogonal to the orthogonalized gradient differences, that is, remove from $f'(z_k)$ the orthogonal projections of $f'(z_k)$ onto the individual normalized gradient differences ;
- iterate process with k set to k+1.

1.2. *Special issues*

There are two possible cases of failure for the method as expressed above.

First, the newly computed $f'(y) - f'(x)$ difference may lie within the subspace generated by the previous differences. This leads to a null vector after orthogonalization with respect to these differences.

We will now see that this is not possible under the assumption that the direction d used for the search was a descent one, and that point y has been determined "uphill".

Let us pose the hypothesis that the newly computed $f'(y) - f'(x)$ would lie within the subspace generated by the previous gradient differences. Since the search direction d has been chosen orthogonal to this subspace, then d would be necessarily orthogonal to $f'(y) - f'(x)$. So we would have :

$$(f'(y) - f'(x)) \cdot d = 0$$

which could be rewritten as :

$$(8) \quad f'(x) \cdot d - f'(y) \cdot d = 0$$

But from the fact that d is a descent direction, we have $f'(x) \cdot d < 0$, and from the fact that y has been found "uphill", we have $f'(y) \cdot d > 0$; therefore we obtain :

$$f'(x) \cdot d - f'(y) \cdot d < 0$$

which is contradictory with (8). So, a newly computed $f'(y) - f'(x)$ cannot lie with the subspace generated by the previous gradient differences.

Second, we will now show that the method cannot generate a direction that is not a descent one.

Due to the fact that the direction has been chosen as the projection of the gradient onto the subspace orthogonal to the subspace generated by previous gradient differences, the new direction is not a descent one if and only if

the gradient lies within the subspace generated by the previous differences. In this case the direction d would vanish, thus preventing the process from being continued.

A vanishing new direction would mean that the gradient at the iteration point is either zero or orthogonal to the subspace where the optimum lies. If the gradient is null then the optimum is achieved. If not, at this iteration point we would have :

$$f'(x) \cdot (x-x^*) = 0$$

But, using (4), we would get :

$$f'(x) \cdot (x-x^*) = G'_1(1, f(x)) = 0$$

which is contradictory with (3). So, the resulting direction as chosen above cannot vanish when used on HIS functions.

1.3. *Special case : quadratic f*

It is interesting to see what becomes of β as defined above when f is a quadratic function, which is obviously a special case of a HIS function. In this case, since the gradient of f is linear, we have :

$$f'(y) = f'(x) + \alpha Ad$$

where A is the Hessian matrix of f , and Ad the matrix-vector product of A by d .

So the expression (7) yielding β can be simplified and rewritten as :

$$(9) \quad \beta = \alpha + \frac{f'(x) \cdot d}{d \cdot Ad}$$

Using the fact that f is quadratic, we can write, using Taylor expansion :

$$f(y) = f(x) + \alpha f'(x) \cdot d + \alpha^2 d \cdot Ad / 2$$

and using $f(y)=f(x)$ we get :

$$\alpha f'(x) \cdot d + \alpha^2 d \cdot Ad / 2 = 0$$

The non trivial solution for α is therefore :

$$\alpha = -2 \frac{f'(x) \cdot d}{d \cdot Ad}$$

Replacing and simplifying in (9) gives :

$$\beta = - \frac{f'(x) \cdot d}{d \cdot Ad} = \alpha / 2$$

which means that, as could be expected, z is midway between x and y , i.e. z is the minimum of f along (D) .

For quadratic functions, at every iteration k , the gradient difference $f'(x_k) - f'(y_k)$ is colinear to Ad_k . Since the new direction d_{k+1} is computed to be orthogonal to the subspace generated by the previous gradient differences, we then have :

$$d_{k+1} \cdot Ad_j = 0, \text{ for } j = 0 \text{ to } k$$

This means that the direction d_{k+1} is conjugate to d_j with respect to A for all $j \leq k$.

Given a sequence of n independent vectors, this property allows building univocally a set of conjugate directions which spans the same subspace than the original vectors.

In both this method and the conjugate gradient algorithm, the initial direction is the opposite to the gradient at the starting point. In both methods also, the new direction is uniquely defined by the fact it is the projection of the gradient at the iteration point onto the subspace orthogonal to the subspace generated by the previous gradient vectors. Hence, by induction, the resulting directions are the same in both methods.

Therefore, when applied to quadratic functions, the method presented in this paper generates the same directions (and thereafter the same iteration points) as the conjugate gradient algorithm.

1.4. Down sized algorithm for huge problems

Increasingly often, modern practical problems involve a huge number of variables. It is therefore interesting to note that this method can be easily modified to accommodate whatever amount of storage is available.

This is accomplished by keeping only the last p gradient differences (where p is some integer less than n), while discarding older differences. Of course, the finite termination does not hold any longer in this case, but it is not a major concern when the method is applied to non HIS functions for which this feature does not hold anyway.

So, the method can be dynamically adjusted to whatever storage is reasonably available, while still remaining a powerful tool.

This feature can be a definite advantage when compared to classic matrix methods such as Huang family methods [3].

Numerical experiments presented in section 3.2 include results for such a down sized algorithm.

2. MINIMIZING NON "HIS" FUNCTIONS

The algorithm developed in section 1 does not explicitly rely on the fact that the function being minimized is a HIS one. Therefore, the method can be used to minimize general nonlinear functions, provided their first derivatives are known. However, some of the results established in section 4 do not hold any longer when applied to non HIS functions, and this must be accounted for.

More specifically, there may not be a point y such that $f(y)=f(x)$ along a descent path starting at x , and the gradient at a newly computed iteration point may lie within the subspace generated by gradient differences. In both cases, the algorithm cannot be continued and the only possibility left is to restart the process from the beginning, i.e. with a search direction set to the gradient. If the process were to fail again at this point, the algorithm should be stopped and a failure be reported.

Note that the method (as exposed in section 1) does not require line searches for a minimum. This peculiarity may sound odd when compared to classical

minimization methods where linear searches for a minimum form the building blocks of the process. However, preliminary experiments showed that there was no special advantage adding an extra search for a minimum at the end of the algorithm, and therefore this feature has been discarded.

By the way, it is interesting to remark that searching for a point y such that $f(y)=f(x)$ is faster than searching for a minimum when using a dichotomous line search : in the first instance, the worst case rate of convergence is 0.5, in the second instance ("golden" search for a minimum), the rate is only 0.618.

Now, our algorithm can be stated more precisely as follows :

0. set x_0 = some initial guess
1. note: starting direction is opposite of gradient
 set $d_0 = -f'(x_0)$
 note: k is the iteration number, $0 \leq k < n$
 set $k = 0$
2. note: ε = some predefined threshold
 if $|f'(x_k)| < \varepsilon$ then go to 7
3. search $\alpha_k > 0$ such that $f(x_k + \alpha_k d_k) = f(x_k)$
 if not such α_k then go to 6
4. note: determination of next iteration point
 set $y_k = x_k + \alpha_k d_k$
 let $\beta_k = \alpha_k f'(y_k) \cdot d_k / (f'(y_k) - f'(x_k)) \cdot d_k$
 let $x_{k+1} = x_k + \beta_k d_k$
 note: orthogonalization of the gradient difference
 let $\delta_k = f'(y_k) - f'(x_k)$
 for $j=0$ to $k-1$ let $\delta_k = \delta_k - (\delta_k \cdot \delta_j) \delta_j$
 let $\delta_k = \delta_k / |\delta_k|$
 note: orthogonalization of the new direction of search
 let $d_{k+1} = -f'(x_{k+1})$

```

for j=0 to k let  $d_{k+1} = d_{k+1} - (d_{k+1} \cdot \delta_j) \delta_j$ 
if  $|d_{k+1}| < \varepsilon$  then go to 6
5. let  $d_{k+1} = d_{k+1} / |d_{k+1}|$ 
set  $k = k + 1$ 

if  $k \geq n$  then go to 1, else go to 2
6. if  $k=0$  then stop 'Failure to converge'

set  $x_0 = x_k$ 
go to 1
7. display 'Minimum reached'
display  $x^k$  and other relevant information
stop

```

3. NUMERICAL EXPERIMENTS

3.1. Overview

In section 3.2, we will report some numerical experiments, with both the full version and a down sized version of this algorithm. The down sized method is obtained by keeping only the last four gradient differences as mentioned in 1.2.2.

The numbers of iterations they required to achieve convergence are compared to those of the well known BFGS [2] method and Polak-Ribière version of the conjugate gradient [4] method.

These experiments have been conducted under the following assumptions :

- the convergence was considered to be obtained when the optimum value is achieved with a precision better than 10^{-10} ;
- in any case of trouble (non decreasing function over a line search, failure to generate a new direction, etc.), the failing method was restarted from the beginning, using the current point of iteration as the new starting point, throwing away any remembered information ; the failure was considered definitive when this restart attempt also failed ;

- all line searches have been performed with maximum precision (10^{-16}) to avoid roundoff effects.

A little difficulty arises when comparing the performances of the methods, because classic algorithms (CG and BFGS) use only one gradient evaluation per step instead of two in our method. To circumvent this, two ranking criteria were used. The first one was the number of iterations used to achieve the convergence : it is well suited to problems with small numbers of variables, where the gradient evaluation cost is likely to be negligible when compared to line search cost. The second criterion was the number of gradient evaluations necessary to achieve the convergence : it is well suited to problems with a great number of variables, where line search cost can probably be neglected when compared to gradient evaluation cost. This allows a potential user of the method to put together his own criterion from these.

The following abbreviations have been employed thereafter :

BFGSM : Broyden Fletcher Goldfarb Shanno Method [2]
 CGM : Conjugate Gradient Method [4]
 4HISM : 4 directions Homotetic Isovalue Surface Method
 FHISM : Full Homothetic Isovalue Surface Method

3.2. Test Problems

3.2.1. Huang and Levy Quasi-Quadratic Function

number of variables: $n = 4$

function: $f(x) = [Q(x)]^p$

where: $Q(x) = \frac{1}{2}xAx + bx + c$

$$A = \begin{bmatrix} 4.5 & 7 & 3.5 & 3 \\ 7 & 14 & 9 & 8 \\ 3.5 & 9 & 8.5 & 5 \\ 3 & 8 & 5 & 7 \end{bmatrix} \quad b = \begin{bmatrix} -0.5 \\ -1.0 \\ -1.5 \\ 0 \end{bmatrix} \quad c = 0.25$$

parameter: $p = 0.3$

starting point: $x_i = 1, i = 1 \text{ to } n$

solution $x_i^* = 0, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	4	4	45	80
Iteration ranking	1	1	3	4
Gradient ranking	1	1	3	4

3.2.2. Huang and Levy Quasi-Quadratic Function

number of variables: $n = 4$

function: $f(x) = [Q(x)]^p$

where: $Q(x) = \frac{1}{2}xAx + bx + c$

$$A = \begin{bmatrix} 4.5 & 7 & 3.5 & 3 \\ 7 & 14 & 9 & 8 \\ 3.5 & 9 & 8.5 & 5 \\ 3 & 8 & 5 & 7 \end{bmatrix} \quad b = \begin{bmatrix} -0.5 \\ -1.0 \\ -1.5 \\ 0 \end{bmatrix} \quad c = 0.25$$

parameter: $p = 0.5$

starting point: $x_i = 1, i = 1 \text{ to } n$

solution $x^* = 0, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	4	4	29	40
Iteration ranking	1	1	3	4
Gradient ranking	1	1	3	4

3.2.3. Huang and Levy Quasi-Quadratic Function

number of variables: $n = 4$

function: $f(x) = [Q(x)]^p$

where: $Q(x) = \frac{1}{2}xAx + bx + c$

$$A = \begin{bmatrix} 4.5 & 7 & 3.5 & 3 \\ 7 & 14 & 9 & 8 \\ 3.5 & 9 & 8.5 & 5 \\ 3 & 8 & 5 & 7 \end{bmatrix} \quad b = \begin{bmatrix} -0.5 \\ -1.0 \\ -1.5 \\ 0 \end{bmatrix} \quad c = 0.25$$

parameter: $p = 1.0$

starting point: $x_i = 1, i = 1 \text{ to } n$

solution $x^* = 0, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	4	4	4	5
Iteration ranking	1	1	1	4
Gradient ranking	3	3	1	2

3.2.4. Huang and Levy Quasi-Quadratic Function

number of variables: $n = 4$

function: $f(x) = [Q(x)]^p$

where: $Q(x) = \frac{1}{2}xAx + bx + c$

$$A = \begin{bmatrix} 4.5 & 7 & 3.5 & 3 \\ 7 & 14 & 9 & 8 \\ 3.5 & 9 & 8.5 & 5 \\ 3 & 8 & 5 & 7 \end{bmatrix} \quad b = \begin{bmatrix} -0.5 \\ -1.0 \\ -1.5 \\ 0 \end{bmatrix} \quad c = 0.25$$

parameter: $p = 2.0$

starting point: $x_i = 1, i = 1 \text{ to } n$

solution $x^* = 0, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	4	4	18	35
Iteration ranking	1	1	3	4
Gradient ranking	1	1	3	4

3.2.5. Huang and Levy Quasi-Quadratic Function

number of variables: $n = 4$

function: $f(x) = [Q(x)]^p$

where: $Q(x) = \frac{1}{2}xAx + bx + c$

$$A = \begin{bmatrix} 4.5 & 7 & 3.5 & 3 \\ 7 & 14 & 9 & 8 \\ 3.5 & 9 & 8.5 & 5 \\ 3 & 8 & 5 & 7 \end{bmatrix} \quad b = \begin{bmatrix} -0.5 \\ -1.0 \\ -1.5 \\ 0 \end{bmatrix} \quad c = 0.25$$

parameter: $p = 3.0$

starting point: $x_i = 1, i = 1 \text{ to } n$

solution $x^* = 0, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	4	4	14	28
Iteration ranking	1	1	3	4
Gradient ranking	1	1	3	4

3.2.6. Huang and Levy Quasi-Quadratic Function

number of variables: $n = 4$

function: $f(x) = [Q(x)]^p$

where: $Q(x) = \frac{1}{2}xAx + bx + c$

$$A = \begin{bmatrix} 4.5 & 7 & 3.5 & 3 \\ 7 & 14 & 9 & 8 \\ 3.5 & 9 & 8.5 & 5 \\ 3 & 8 & 5 & 7 \end{bmatrix} \quad b = \begin{bmatrix} -0.5 \\ -1.0 \\ -1.5 \\ 0 \end{bmatrix} \quad c = 0.25$$

parameter: $p = 4.0$

starting point: $x_i = 1, i = 1 \text{ to } n$

solution $x^* = 0, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	4	4	10	24
Iteration ranking	1	1	3	4
Gradient ranking	1	1	3	4

3.2.7. Hilbert Quasi-Quadratic Function

number of variables: $n = 5$

function: $f(x) = (\frac{1}{2}xHx)^2$

where: $H_{ij} = 1/(i+j-1), i = 1 \text{ to } n, j = 1 \text{ to } n$

starting point: $x_i = -3, i = 1 \text{ to } n$

solution $x^* = 0, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	3	3	8	14
Iteration ranking	1	1	3	4
Gradient ranking	1	1	3	4

3.2.8. Quasi-Quadratic Function

number of variables: $n = 10$
 function: $f(x) = \sin(Q(x)) + 1.001 * Q(x)$
 where: $Q(x) = \sum_{i=1}^n i x_i^2$, $i = 1$ to n
 starting point: $x_i = 1$, $i = 1$ to n
 solution: $x^* = 0$, $f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	10	10	16	18
Iteration ranking	1	1	3	4
Gradient ranking	3	3	1	2

3.2.9. Homothetic Isovalue Surfaces Function

number of variables: $n = 10$
 function: $f(x) = \ln(1 + \sum_{i=1}^n i |x_i|^p)$
 parameter: $p = 1.0$
 starting point: $x_i = 1$, $i = 1$ to n
 solution: $x^* = 0$, $f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	30*	10	191	173*
Iteration ranking	3	1	2	4
Gradient ranking	3	1	2	4

* method failed to converge

3.2.10. Homothetic Isovalue Surfaces Function

number of variables: $n = 10$
 function: $f(x) = \ln(1 + \sum_{i=1}^n i |x_i|^p)$
 parameter: $p = 3.0$
 starting point: $x_i = 1$, $i = 1$ to n
 solution: $x^* = 0$, $f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	7	7	22	11
Iteration ranking	1	1	4	3
Gradient ranking	2	2	4	1

3.2.11. Homothetic Isovalue Surfaces Function

number of variables: $n = 10$
function: $f(x) = (\sum_i |x_i|^p)^p$
parameter: $p = 1.0$
starting point: $x_i = 1, i = 1 \text{ to } n$
solution: $x^* = 0, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	29*	10	159	209*
Iteration ranking	3	1	2	4
Gradient ranking	3	1	2	4

* method failed to converge

3.2.12. Homothetic Isovalue Surfaces Function

number of variables: $n = 10$
function: $f(x) = (\sum_i |x_i|^p)^p$
parameter: $p = 3.0$
starting point: $x_i = 1, i = 1 \text{ to } n$
solution: $x^* = 0, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	4	4	21	6
Iteration ranking	1	1	4	3
Gradient ranking	2	2	4	1

3.2.13. Homothetic Isovalue Surfaces Function

number of variables: $n = 10$
function: $f(x) = \sin(F(x)) + 1.001F(x)$
 $F(x) = \sum_i |x_i|^p$
parameter: $p = 1.0$

starting point: $x_i = 1, i = 1 \text{ to } n$
 solution $x^* = 0, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	20*	11	186	191*
Iteration ranking	3	1	2	4
Gradient ranking	3	1	2	4

* method failed to converge

3.2.14. Homothetic Isovalue Surfaces Function

number of variables: $n = 10$
 function: $f(x) = \sin(F(x)) + 1.001F(x)$
 $F(x) = \sum_i |x_i|^p$
 parameter: $p = 3.0$
 starting point: $x_i = 1, i = 1 \text{ to } n$
 solution $x^* = 0, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	8	7	26	9
Iteration ranking	2	1	4	3
Gradient ranking	3	2	4	1

3.2.15. Fletcher-Powell Helical Function

number of variables: $n = 3$
 function: $f(x) = 100[(x_3 - 10\theta)^2 + (r-1)^2] + x^2$
 where :

$$2\pi\theta = \begin{cases} \arctan(x_2/x_1) & , x_1 > 0 \\ \arctan(x_2/x_1) + \pi & , x_1 < 0 \end{cases}$$

$$r = \frac{(x_1^2 + x_2^2)^{1/2}}{2}$$

starting point: $x_0 = (-1, 0, 0)$
 solution $x^* = (1, 0, 0), f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	24	24	24	30
Iteration ranking	1	1	1	4
Gradient ranking	3	3	1	2

3.2.16. Rosenbrock's 2D Function

number of variables: $n = 2$

function: $f(x, y) = (x-1)^2 + 100(y-x^2)^2$

starting point: $(x_0, y_0) = (0, 0)$

solution $(x^*, y^*) = (1, 1), f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	18	18	21	21
Iteration ranking	1	1	3	3
Gradient ranking	3	3	1	1

3.2.17. Rosenbrock's 4D Function

number of variables: $n = 4$

function: $f(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 +$
 $90(x_3^2 - x_4)^2 + (1 - x_3)^2 +$
 $10.1[(x_2 - 1)^2 + (x_4 - 1)^2] +$
 $19.8(x_2 - 1)(x_4 - 1)$

starting point: $x_0 = (-3, -1, -3, -1)$

solution $x^* = (1, 1, 1, 1), f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	59	59	38	65
Iteration ranking	2	2	1	4
Gradient ranking	3	3	1	2

3.2.18. Generalized Rosenbrock's 4D Function

number of variables: $n = 4$

function:
$$f(x) = \sum_{i=1}^{n/4} \{ 100(x_{4i-3}^2 - x_{4i-2})^2 + (1-x_{4i-3})^2 + 90(x_{4i-1}^2 - x_{4i})^2 + (1-x_{4i-1})^2 + 10.1[(x_{4i-2}-1)^2 + (x_{4i}-1)^2] + 19.8(x_{4i-2}-1)(x_{4i}-1) \}$$

starting point: $x_i = -3, i = 1 \text{ to } n$

solution $x_i^* = 1, i = 1 \text{ to } n, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	57	57	46	84
Iteration ranking	2	2	1	4
Gradient ranking	3	3	1	2

3.2.19. Generalized Rosenbrock's 4D Function

number of variables: $n = 20$

function:
$$f(x) = \sum_{i=1}^{n/4} \{ 100(x_{4i-3}^2 - x_{4i-2})^2 + (1-x_{4i-3})^2 + 90(x_{4i-1}^2 - x_{4i})^2 + (1-x_{4i-1})^2 + 10.1[(x_{4i-2}-1)^2 + (x_{4i}-1)^2] + 19.8(x_{4i-2}-1)(x_{4i}-1) \}$$

starting point: $x_i = -3, i = 1 \text{ to } n$

solution $x_i^* = 1, i = 1 \text{ to } n, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	73	71	90	64
Iteration ranking	3	2	4	1
Gradient ranking	4	3	2	1

3.2.20. Generalized Rosenbrock's 4D Function

number of variables: $n = 80$

function:
$$f(x) = \sum_{i=1}^{n/4} \{ 100(x_{4i-3}^2 - x_{4i-2})^2 + (1-x_{4i-3})^2 + 90(x_{4i-1}^2 - x_{4i})^2 + (1-x_{4i-1})^2 + 10.1[(x_{4i-2}-1)^2 + (x_{4i}-1)^2] + 19.8(x_{4i-2}-1)(x_{4i}-1) \}$$

starting point: $x_i = -3, i = 1 \text{ to } n$

solution $x_i^* = 1, i = 1 \text{ to } n, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	68	68	209	87
Iteration ranking	1	1	4	3
Gradient ranking	2	2	4	1

3.2.21. Generalized Rosenbrock's 4D Function

number of variables: $n = 200$

function:
$$f(x) = \sum_{i=1}^{n/4} \{ 100(x_{4i-3}^2 - x_{4i-2})^2 + (1-x_{4i-3})^2 + 90(x_{4i-1}^2 - x_{4i})^2 + (1-x_{4i-1})^2 + 10.1[(x_{4i-2}-1)^2 + (x_{4i}-1)^2] + 19.8(x_{4i-2}-1)(x_{4i}-1) \}$$

starting point: $x_i = -3, i = 1 \text{ to } n$

solution $x_i^* = 1, i = 1 \text{ to } n, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	79	64	416	120
Iteration ranking	2	1	4	3
Gradient ranking	3	2	4	1

3.2.22. Generalized Rosenbrock's 4D Function

number of variables: $n = 500$

function:
$$f(x) = \sum_{i=1}^{n/4} \{ 100(x_{4i-3}^2 - x_{4i-2})^2 + (1-x_{4i-3})^2 + 90(x_{4i-1}^2 - x_{4i})^2 + (1-x_{4i-1})^2 + 10.1[(x_{4i-2}-1)^2 + (x_{4i}-1)^2] + 19.8(x_{4i-2}-1)(x_{4i}-1) \}$$

starting point: $x_i = -3, i = 1 \text{ to } n$

solution $x_i^* = 1, i = 1 \text{ to } n, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	56	0 ^a	0 ^a	210
Iteration ranking	1	3	3	2
Gradient ranking	1	3	3	2

^a method failed because of insufficient memory

3.2.23. Powell's Quartic Function

number of variables: $n = 4$

function:
$$f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 -$$

starting point: $x_0 = (3, -1, 0, 1)$

solution $x^* = 0, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	23	23	22	67
Iteration ranking	2	2	1	4
Gradient ranking	2	2	1	4

3.2.24. Powell's Extended Quartic Function

number of variables: $n = 20$

function:
$$f(x) = \sum_{i=1}^{n/4} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4]$$

starting point: $(x_{4i-3}, x_{4i-2}, x_{4i-1}, x_{4i}) = (3, -1, 0, 3), i = 1 \text{ to } n/4$

solution $x^* = 0, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	30	28	30	83
Iteration ranking	2	1	2	4
Gradient ranking	3	2	1	4

3.2.25. Powell's Extended Quartic Function

number of variables: $n = 80$

function:
$$f(x) = \sum_{i=1}^{n/4} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4]$$

starting point: $(x_{4i-3}, x_{4i-2}, x_{4i-1}, x_{4i}) = (3, -1, 0, 3), i = 1 \text{ to } n/4$

solution $x^* = 0, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	31	34	33	168
Iteration ranking	1	3	2	4
Gradient ranking	2	3	1	4

3.2.26. Powell's Extended Quartic Function

number of variables: $n = 200$

function:
$$f(x) = \sum_{i=1}^{n/4} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4]$$

starting point: $(x_{4i-3}, x_{4i-2}, x_{4i-1}, x_{4i}) = (3, -1, 0, 3), i = 1 \text{ to } n/4$

solution $x^* = 0, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	33	31	80	241
Iteration ranking	2	1	3	4
Gradient ranking	2	1	3	4

3.2.27. Powell's Extended Quartic Function

number of variables: $n = 500$

$$\text{function: } f(x) = \sum_{i=1}^{n/4} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4]$$

starting point: $(x_{4i-3}, x_{4i-2}, x_{4i-1}, x_{4i}) = (3, -1, 0, 3), i = 1 \text{ to } n/4$

solution $x^* = 0, f^* = 0$

	4HISM	FHISM	BFGSM	CGM
Iteration number	41	0 ^a	0 ^a	509
Iteration ranking	1	3	3	2
Gradient ranking	1	3	3	2

^a method failed because of insufficient memory

3.2.28. Statistics

The following tables show the number of times a given method obtained a given rank over the 27 test problems above. Every number is followed by the corresponding percentage in parenthesis. Note that percentages in a line may add up to more than 100% because there can be more than one method having a given rank in case of a tie on a test problem.

Iteration Number Ranking

Rank	4HISM	FHISM	BFGSM	CGM
1	16 (59%)	20 (74%)	5 (19%)	1 (4%)
2	7 (26%)	4 (15%)	5 (19%)	2 (7%)
3	4 (15%)	3 (11%)	11 (41%)	6 (22%)
4	0 (0%)	0 (0%)	6 (22%)	18 (67%)

Gradient Number Ranking

Rank	4HISM	FHISM	BFGSM	CGM
1	8 (30%)	10 (37%)	9 (33%)	7 (26%)
2	6 (22%)	7 (26%)	4 (15%)	7 (26%)
3	12 (44%)	10 (37%)	9 (33%)	0 (0%)
4	1 (4%)	0 (0%)	5 (19%)	13 (48%)

The following table shows the number of failures for every method.

	4HISM	FHISM	BFGSM	CGM
Failure number	3 (11%)	2 (7%)	2 (7%)	3 (11%)

3.3. Discussion

At first glance, the major outcome of the above experiments is that on the problems tested, the two versions of our algorithm performed similarly, despite keeping only four directions in the down sized algorithm.

For the iteration criterion, the down sized version and the full version of our method respectively occupy 59% and 74% of first ranks, which is clearly superior to other methods.

For the gradient number criterion, the results are more balanced : for first rank, the full version of our algorithm comes first (37%), but is closely followed by the BFGS method (33%), itself followed by the down sized version of our algorithm (30%) and finally the conjugate gradient method (26%). This is due to the fact that our method has to undergo the severe handicap of two gradient evaluations per iteration step, instead of one for the BFGS method or the CG algorithm. Similar results hold for the lower ranks.

However, it is worthwhile to note that on the two largest (500 variables) test problems (Rosenbrock and Powell's extended functions), our down sized method still wins the first rank despite the above mentioned handicap. This is partly due to the fact that for problems of such a size, matrix method (BFGS and full HISM) are unusable. However, this stresses the point that the down sized version of our method seems to be very effective compared to the CG algorithm, the only alternative method available for very big problems.

Hence, these experiments allow us to believe that the method presented here may prove even more interesting for larger problems like those arising from structural design optimization, and that it should be rewarding to investigate further its use on such large practical problems.

4. MATHEMATICAL STUDY

In this section, we will review some mathematical properties of the functions verifying (1).

4.1. Homothetic isovalue surfaces

Let F be a function verifying (1).

It is easy to show that the isovalue surfaces of F are homothetic to one another with respect to the origin of coordinates.

Let (S_1) be the isovalue surface of F defined by :

$$(10) \quad F(X) = \alpha, \quad \alpha \in \mathbb{R}$$

and (S_2) be the isovalue surface of F defined by :

$$F(Y) = \beta, \quad \beta \in \mathbb{R}$$

We can assume without loss of generality that $\alpha \geq \beta \geq F(0)$.

Let us consider the function $f : \mathbb{R} \rightarrow \mathbb{R}$ defined as follow for a given X :

$$f(\mu) = F(\mu X)$$

$f(\mu)$ varies from $f(0)=F(0)$ to $f(1)=\alpha$ when μ varies from 0 to 1. Since F is continuous, so is f . Therefore, since $f(0) \leq \beta \leq f(1)$, there exists some $\mu \in [0,1]$ such that $f(\mu)=\beta$, i.e. $F(\mu X)=\beta$.

Using (1), we have :

$$F(\mu X) = G(\mu, F(X)) = G(\mu, \alpha) = \beta$$

According to (1), for a fixed α the function defined by $G_{\alpha}(\mu) \equiv G(\mu, \alpha)$ is a continuous, monotonous and increasing function of μ in $[0,1]$, so it has an inverse function G_{α}^{-1} such that :

$$\mu = G_{\alpha}^{-1}(\beta)$$

Therefore, μ does not depend on X , which shows that (S1) and (S2) are homothetic.

Reciprocally, any surface homothetic to an isovalue surface of F is also an isovalue surface of F . Consider the isovalue surface (S1) defined by (10). Then the surface (S2) defined by :

$$Y = \mu X, \quad \text{for some fixed } \mu \text{ in } \mathbb{R}^{*+} \text{ and where } X \text{ verifies (10)}$$

is homothetic to (S1) with respect to the origin of coordinates. Using (1), we get :

$$F(Y) = F(\mu X) = G(\mu, F(X)) = G(\mu, \alpha) = \text{some constant in } \mathbb{R}$$

So, (S2) is also an isovalue surface of F .

The fact that isovalue surfaces are homothetic to one another is the main characterization of functions verifying (1). So, such "Homothetic Isovalue Surfaces" functions defined by (1) will be described as "HIS" functions.

4.2. Fundamental property of the gradient of a HIS function

Differentiating definition (1) with respect to μ yields :

$$F'(\mu X) \cdot X = G_1(\mu, F(X))$$

where F' stands for the gradient of F and G_1 for the derivative with respect to the first variable of G ,

and \cdot denotes the scalar product of two vectors $\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$.

Taking $\mu=1$ in equation above yields :

$$(11) \quad F'(X) \cdot X = G_1(1, F(X))$$

on an isovalue surface (S) defined by $F(X)=\alpha$, we then have :

$$F'(X) \cdot X = G_1(1, \alpha) = \text{constant in } \mathbb{R}^{*+}$$

This means that the scalar product $F'(X) \cdot X$ is constant on any isovalue surface of F . This is the fundamental property which has been used previously in section 2 to minimize exactly HIS functions.

Reciprocally, assume we have some function $F : \mathbb{R}^n \rightarrow \mathbb{R}$ with continuous derivatives, such that :

$$(12) \quad F'(X) \cdot X = h(F(X)), \quad \forall X \in \mathbb{R}^n$$

where $h : \mathbb{R} \rightarrow \mathbb{R}^{*+}$

Rewriting (12) for μX , $\mu \in \mathbb{R}^{*+}$, we get :

$$F'(\mu X) \cdot (\mu X) = h(F(\mu X))$$

that is :

$$(13) \quad \frac{F'(\mu X) \cdot X}{h(F(\mu X))} = 1/\mu$$

$h(y)$ being defined and positive for all $y > F(0)$ implies that $1/h(y)$ is also defined and positive for all $y > F(0)$. Therefore, there exists a function $H : \mathbb{R} \rightarrow \mathbb{R}$ such that :

$$H'(y) = 1/h(y), \quad y > F(0)$$

where H' stands for the first derivative of H .

Thus, (13) can be rewritten as follows :

$$\frac{dH(F(\mu X))}{d\mu} = 1/\mu$$

Integrating this, we obtain :

$$H(F(\mu X)) = \ln(\mu) + c(X)$$

where $\ln : \mathbb{R} \rightarrow \mathbb{R}$ is the Napierian logarithm,

and $c : \mathbb{R}^n \rightarrow \mathbb{R}$ is some constant with respect to μ .

For $\mu=1$, we get :

$$H(F(X)) = c(X)$$

So, we finally obtain :

$$(14) \quad H(F(\mu X)) = \ln(\mu) + H(F(X))$$

From the fact that $H'(y)$ is continuous and positive, we infer that H has a inverse function $H^{-1} : \mathbb{R} \rightarrow \mathbb{R}$; applying H^{-1} to both sides of (14), we obtain :

$$F(\mu X) = H^{-1}(\ln(\mu) + H(F(X)))$$

Defining $G(x, y) \equiv H^{-1}(\ln(x) + H(y))$ for $(x, y) \in \mathbb{R}^{*+} \times \mathbb{R}$, we get :

$$F(\mu X) = G(\mu, F(X))$$

which is precisely the form (1). From this and the facts that :

$$\begin{aligned} G'_1(\mu, \alpha) &= 1/H'(H^{-1}(\ln(\mu) + H(\alpha))) \cdot 1/\mu \\ &= h(H^{-1}(\ln(\mu) + H(\alpha))) / \mu \end{aligned}$$

$$\begin{aligned} \text{and } G'_2(\mu, \alpha) &= 1/H'(H^{-1}(\ln(\mu) + H(\alpha))) \cdot H'(\alpha) \\ &= h(H^{-1}(\ln(\mu) + H(\alpha))) / h(\alpha) \end{aligned}$$

are both continuous and positive for all μ, α , we can conclude that any function verifying (12) is a HIS function.

4.3. Relation between the gradient and the Hessian matrix

Differentiating (11) with respect to X yields :

$$F''(X) * X + F'(X) = G''_{12}(1, F(X)) F'(X)$$

where $F'' : \mathbb{R}^n \rightarrow \mathbb{R}^{n^2}$ is the Hessian matrix of F ,

$*$ denotes the matrix-vector product,

and G''_{12} is the second derivative of G with respect to the first and the second variable in turn.

This can be rewritten as :

$$F''(X) * X = [G''_{12}(1, F(X)) - 1] F'(X)$$

which expresses that everywhere, the gradient is colinear to the product of the Hessian matrix by the gradient.

4.4. *Existence of an $\alpha > 0$ such that $F(X + \alpha D) = F(X)$ for all X , D being a descent direction*

Let X be some point of \mathbb{R}^n , and D some descent direction for F that defines a line passing through X . We will show now that if F is a HIS function, there always exists an $\alpha \in \mathbb{R}^{*+}$ such that :

$$F(X + \alpha D) = F(X)$$

First, we will show that there exists some $\beta \in \mathbb{R}^{*+}$ such that :

$$F(X + \beta D) \geq F(X)$$

The proof is by refutation. Assume there were to be no such β for some given X, D . Then, we would have :

$$F(X + \beta D) < F(X) \quad \forall \beta \in \mathbb{R}^{*+}$$

Let us consider the following points :

$$Y = X/\beta$$

$$\text{and } Z = (X+\beta D)/\beta = Y + D$$

The values of F at these points would be :

$$F(Y) = F(X/\beta) = G(1/\beta, F(X))$$

$$\text{and } F(Z) = F((X+\beta D)/\beta) = G(1/\beta, F(X+\beta D))$$

$G(\mu, \beta)$ is an increasing function of β for all $\mu > 0$, $\beta > 0$. So $F(X+\beta D) < F(X)$ for all $\beta > 0$ would then imply that $F(Z) < F(Y)$ for all $\beta > 0$. Therefore, if we were to make β go to infinity, then Y would converge to 0 and Z would converge to D . F being continuous, then $F(Y)$ would converge to $F(0)$. But $F(Y) > F(Z) \geq F(0)$ would imply in turn that $F(Z)$ would also converge to $F(0)$. Since F is continuous, this would imply that $F(D) = F(0)$. Since D is a descent direction, it is not the null vector and therefore F would have two distinct minima 0 and D , which is contradictory with (1). Hence, for any descent direction D from a point X , there exists some $\beta > 0$ such that $F(X+\beta D) \geq F(X)$.

So, there exist both a real $\beta > 0$ such that $F(X+\beta D) \geq F(X)$ and a real $\varepsilon > 0$ such that $F(X+\varepsilon D) < F(X)$. Since F is continuous, this implies that there is some real $\beta > \alpha > \varepsilon > 0$ such that $F(X+\alpha D) = F(X)$, whenever F is a HIS function and D is a descent direction from X .

4.5. Building a HIS function

Let there be a hypersurface defined by the implicit equation :

$$S(X) = 0, X \in \mathbb{R}^n$$

such that for any given $Y \in \mathbb{R}^n$, $Y \neq 0$, the equation :

$$S(\tau Y) = 0$$

has a unique solution $\tau \in \mathbb{R}^{*+}$.

Let there be some function $K : \mathbb{R}^{*+} \rightarrow \mathbb{R}$ having a continuous first derivative K' such that $K'(\mu) > 0$ for all $\mu \in \mathbb{R}^{*+}$; then we can define a function $F : \mathbb{R}^n \rightarrow \mathbb{R}$ by :

$$F(X) = K(\tau),$$

where τ is the unique solution of $S(\tau X)=0$.

Rewriting the previous equation for μX yields :

$$F(\mu X) = K(\sigma),$$

where σ is the unique solution of $S(\sigma \mu X)=0$.

Therefore $\sigma=\tau/\mu$, where τ is the unique solution of $S(\tau X)=0$.

Since K has a continuous positive derivative, it has an inverse function K^{-1} $\mathbb{R} \rightarrow \mathbb{R}^{*+}$ such that :

$$K^{-1}[K(\tau)] = \tau \quad \forall \tau \in \mathbb{R}^{*+}.$$

So we have : $\tau = K^{-1}[F(X)]$, and therefore :

$$F(\mu X) = K(K^{-1}[F(X)]/\mu)$$

Let $G : \mathbb{R}^n \rightarrow \mathbb{R}$ be defined by :

$$G(\mu, \alpha) = K(K^{-1}[\alpha]/\mu) \quad \forall \mu \in \mathbb{R}^{*+}, \forall \alpha \in \mathbb{R}, \alpha > F(0).$$

Then $F(\mu X) = G(\mu, F(X))$ and :

$$G'_1(\mu, \alpha) = K'(K^{-1}[\alpha]/\mu) K^{-1}[\alpha]/\mu^2$$

is positive from the fact that $K'(\mu) > 0$ wherever it is defined, and that K^{-1} takes its values in \mathbb{R}^{*+} . Similarly :

$$G'_2(\mu, \alpha) = K'(K^{-1}[\alpha]/\mu) (K^{-1}[\alpha])' / \mu =$$

$$K'(K^{-1}[\alpha]/\mu) / K'(K^{-1}[\alpha]) / \mu$$

is positive from the fact that $K'(\mu) > 0$ wherever it is defined and $\mu > 0$.

So, G fulfils the requirements of (1) and thereafter, F is a HIS function. In other terms, this means that we can build a HIS function given an arbitrary hypersurface containing the origin and an arbitrary positive real-valued increasing function. This can be regarded as another way to define the rather broad class of HIS functions.

4.6. Relation to homogeneous functions

A function $F : \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be homogeneous of degree p if it verifies :

$$F(\mu X) = \mu^p F(X) \quad \text{for all } \mu \in \mathbb{R} \text{ and } X \in \mathbb{R}^n$$

Obviously, such a function is a HIS function for which G is defined by :

$$G(\mu, Y) = \mu^p Y$$

Conversely, it is interesting to note that there is a close relationship between HIS functions and homogeneous ones : a HIS function can be transformed into an homogeneous function through nonlinear scaling.

Let $h : \mathbb{R} \rightarrow \mathbb{R}$ be a function with a positive first derivative, and let $K : \mathbb{R}^n \rightarrow \mathbb{R}$ be defined by :

$$K(X) = h(F(X))$$

Such a function K is said to be a nonlinear (if h is nonlinear) scaling of F .

Substituting μX for X yields :

$$K(\mu X) = h(F(\mu X))$$

and differentiating with respect to μ and taking $\mu=1$ yields :

$$K'(X) \cdot X = h'(F(X)) F'(X) \cdot X$$

From (11) then we have :

$$K'(X) \cdot X = h'(F(X)) G'(1, F(X))$$

We can now choose h such that K is homogeneous of degree $p \in \mathbb{R}$; all we need is that :

$$K'(X) \cdot X = pK(X) \text{ for all } X \in \mathbb{R}^n$$

This means that :

$$h'(F(X))G'(1, F(X)) = ph(F(X))$$

If by Y we denote $F(X)$ we get :

$$\frac{h'(Y)}{h(Y)} = \frac{p}{G'(1, Y)}$$

which can be integrated to :

$$\ln(h(Y)) = p \int \frac{dY}{G'(1, Y)}$$

So, h is defined by :

$$h(Y) = \exp\left(p \int \frac{dY}{G'(1, Y)}\right) \text{ for all } Y \in \mathbb{R},$$

and for this choice of h , K is a HIS function.

ACKNOWLEDGMENTS

The author thanks Professor W. BOCQUET for having made this work possible in his laboratory.

The author also thanks Professor P. LAURENT for his careful reading of the manuscript and his helpful suggestions, and Mlle MAGNOUX for her invaluable help in bibliographic database searching.

REFERENCES

- [1] B. BRUET, "A Corrected-Gradient Method that makes Gradient Optimization Algorithms Invariant to any Non-Linear Scaling" *Engineering Optimization*, 1980, Vol. 5, pp. 13-18
- [2] M. AVRIEL, "Nonlinear Programming: Analysis and Methods", Prentice-Hall, Series in Automatic Computation, 1976, ISBN 0-13-623603-0
- [3] H. Y. HUANG, "Unified Approach to Quadratically Convergent Algorithms for Function Minimization", *JOTA*, 5, (6), (1970)
- [4] E. POLAK et G. RIBIERE, "Note sur la Convergence de Méthodes de Directions Conjuguées" *R.I.R.O.* 3ème année, (16), pp. 35-45, (1969)
-