SRC TR 85-37

A Superlinearly Convergent Method of Feasible Directions for Optimization Problems Arising in the Design of Engineering Systems

By

E.R. Panier and A.L. Tits

A Superlinearly Convergent Method of Feasible Directions for Optimization Problems Arising in the Design of Engineering Systems[1]

by

Eliane R. Panier and *André L. Tits*

Electrical Engineering Department
and Systems Research Center
University of Maryland
College Park, MD 20742.

Abstract.

Optimization problems arising from engineering design problems often involve the solution of one or several *constrained minimax* optimization problems. It is sometimes crucial that all iterates constructed when solving such problems satisfy a given set of 'hard' inequality constraints, and generally desirable that the (maximum) objective function value improve at each iteration. In this paper, we propose an algorithm of the sequential quadratic programming (SQP) type that enjoys such properties. This algorithm is inspired from an algorithm recently proposed for the solution of single objective constrained optimization problems. Preliminary numerical results are very promising.

Please address all correspondence to A.L. Tits

[1]This research was supported by the National Science Foundation, grants No. ENG-84-51515, DMC-84-20740 and OIR-85-00108.

1. Introduction.

Many problems encountered in the design of engineering systems can be expressed in the form of a multiobjective optimization problem

$$(P_0)\quad \begin{cases} \min\{f_1(x),\cdots f_l(x)\} \\ \text{s.t.}\ \ g_j(x)\le 0\ \ j=1,\cdots m \end{cases}$$

where $f_i:\mathbb{R}^n \to \mathbb{R}$, $i=1,\cdots l$ and $g_j:\mathbb{R}^n \to \mathbb{R}$, $j=1,\cdots m$, are smooth functions. Usually, a design parameters vector x minimizing all the functions f_i, $i=1,\cdots l$ over the feasible domain does not exist. Problems of this type have been widely studied in the literature (see [1] for an excellent survey). Typical approaches make use of a real-valued preference function $u(f_1,\cdots f_l)$ expressing how satisfied the user is with some given values of the objective functions. The search for an 'optimal' design parameter vector then amounts to solving a standard optimization problem of the form

$$\begin{cases} \max\ u\,\{f_1(x),\cdots f_l(x)\} \\ \text{s.t.}\ \ g_j(x)\le 0\ \ j=1,\cdots m. \end{cases}$$

An important shortcoming of this type of approach is that a preference function is generally difficult to express *a priori*. Thus, typically, tradeoff between the various objective functions must be explored interactively. However, such tradeoff exploration can meaningfully take place only once some 'hard' constraints are satisfied. These constraints, expressing hard design specifications such as stability or physical realizability, are such that their violation would result in a design of no practical value. Since typically, in a design environment, functions evaluations call for computationally expensive system simulations, it is essentially required that hard constraints be satisfied at each iteration. To help tradeoff exploration, it is also desirable that the design obtained after each iteration improve on the previous one.

The ideas just outlined are essential components of the design methodology proposed in [2] , where the preference function is the 'max' function of suitably scaled objectives. Problem (P_0) then becomes

$$(P) \begin{cases} \min f(x) \\ \text{s.t. } x \in X \end{cases}$$

where $f(x) \triangleq \max\{f_1(x), \cdots f_l(x)\}$ and $X = \{x \text{ s.t. } g_j(x) \leq 0 \ j = 1, \cdots m\}$. Here the f_i's are *scaled* versions of the ones in (P_0) and their scaling is interactively adjusted by the designer (see also [3]).

In this framework, the stipulations pointed out above amount to the requirement that, given $x_0 \in X$, the optimization algorithm construct a sequence $\{x_k\}_{k=0}^{\infty}$ such that, for all k ,

$$x_k \in X$$
$$f(x_{k+1}) \leq f(x_k) \tag{0}$$

where we have assumed, for simplicity of exposition, that all constraints are hard. Algorithms satisfying these requirements are available in the literature (see e.g., [2]). They have been used very successfully in solving engineering design problems arising in diverse application areas [4, 5, 6]. However, they suffer from an important shortcoming in that they are generally slow, as their rate of convergence is at best linear. Superlinearly convergent algorithms for solving (P) have been proposed (see e.g. [7, 8]) but they do not satisfy requirements (0).

In [9] , an algorithm satisfying the requirements just outlined was proposed for the case of a single objective function ($l=1$). This algorithm is of the 'sequential quadratic programming' (SQP) type and, under mild assumptions, exhibits a super-linear rate of convergence. Such an algorithm can be used here if (P) is reformulated as the single objective problem in (x, z)

$$(\tilde{P}) \quad \begin{cases} \min z \\ \text{s.t. } f_i(x) \leq z \quad i=1, \cdots l \\ \quad\; g_j(x) \leq 0 \quad j=1, \cdots m. \end{cases}$$

Clearly, solving this problem using the algorithm in [9] would satisfy our requirements. It turns out however that a simpler algorithm can be shown to also meet these requirements. Presenting this algorithm is the object of this paper.

The balance of the paper is as follows. In section 2, the new algorithm is stated and the differences, in comparison with that obtained by using on $(\tilde{P})$ the algorithm in [9] , are briefly discussed. In section 3, convergence and rate of convergence are analyzed. Conclusions are presented in section 4.

2. The Algorithm.

For a given design parameter vector x and a direction d, we will make use of the linearized function obtained from f at x in the direction d by

$$f(x, d) \triangleq \max_{i=1, \cdots l} \{ f_i(x) + \langle \nabla f_i(x), d \rangle \},$$

and of the Lagrangian function associated with a point x and some multiplier vectors λ and μ

$$L(x,\lambda,\mu) = \sum_i \lambda_i f_i(x) + \sum_j \mu_j g_j(x).$$

The algorithm can now be stated.

Algorithm A.

Parameters.

$\alpha \in (0, \frac{1}{2})$, $\beta \in (0,1)$, $\nu > 2$, $\kappa > 2$, $\tau \in (2,3)$.

Data.

Feasible starting point $x_0 \in X$

Symmetric positive definite matrix $H_0 \in \mathbb{R}^{n \times n}$.

Step 0.

Set $k = 0$.

Step 1. Computation of a search direction.

i) Solve

$$(QP_0) \begin{cases} \min \frac{1}{2} d^T H_k d + \max_{i=1,\cdots l} \{ f_i(x_k) + <\nabla f_i(x_k), d> \} \\ \text{s.t. } g_j(x_k) + <\nabla g_j(x_k), d> \leq 0 \quad j = 1, \cdots m \end{cases}$$

to obtain the auxiliary direction d_k^0.

If $| d_k^0 | = 0$ stop.

ii) Solve

$$(QP) \begin{cases} \min \frac{1}{2} d^T H_k d + \max_{i=1,\cdots l} \{ f_i(x_k) + <\nabla f_i(x_k), d> \} \\ \text{s.t. } g_j(x_k) + <\nabla g_j(x_k), d> \leq - | d_k^0 |^\nu \quad j = 1, \cdots m \end{cases}$$

to obtain the search direction d_k. If (QP) has no solution go to (iv).

Problem (QP) is equivalent to the following quadratic program in d and δ

$$\begin{cases} \min_{d,\delta} \frac{1}{2} d^T H_k d + \delta \\ \text{s.t. } f_i(x_k) + <\nabla f_i(x_k), d> \leq \delta \quad i=1, \cdots l \\ g_j(x_k) + <\nabla g_j(x_k), d> \leq -\mid d_k^0 \mid^\nu \quad j=1, \cdots m. \end{cases}$$

We will denote by λ_k and μ_k the multiplier vectors associated respectively with the problem objectives and constraints.

Set $\theta_k = f(x_k, d_k) - f(x_k)$.

If $\theta_k > \min(-\mid d_k^0 \mid^\kappa, -\mid d_k \mid^\kappa)$, go to (iv).

iii) Compute a correction $\tilde{d}_k$, solution of the quadratic program

$$(\tilde{QP}) \begin{cases} \min_d \frac{1}{2}(\mid d \mid^2) \\ \text{s.t. } f_i(x_k + d_k) + <\nabla f_i(x_k), d) = f_j(x_k + d_k) + <\nabla f_j(x_k), d> \forall\, i, j \in I_k \\ g_j(x_k + d_k) + <\nabla g_j(x_k), d> = -\mid d_k^0 \mid^\tau \quad \forall\, j \in J_k \end{cases}$$

where I_k and J_k are defined as

$$I_k = \{ i \text{ s.t. } f_i(x_k) + <\nabla f_i(x_k), d_k> = f(x_k, d_k) \}$$

$$J_k = \{ j \text{ s.t. } g_j(x_k) + <\nabla g_j(x_k), d_k> = -\mid d_k^0 \mid^\nu \}.$$

If $(\tilde{QP})$ has no solution or if $\mid \tilde{d}_k \mid > \mid d_k \mid$, set $\tilde{d}_k = 0$.

Proceed to step 2.

iv) Compute a first order feasible descent direction d_k.

The requirements on this direction are obvious modifications of those in. [9]

Set $\tilde{d}_k = 0$.

Step 2. Line search.

Compute t_k, the first number t of the sequence $\{1, \beta, \beta^2, \ldots\}$ satisfying

$$f(x_k + td_k + t^2 \tilde{d}_k) \leq f(x_k) + \alpha t \theta_k \qquad (1)$$

$$g_j(x_k + td_k + t^2 \tilde{d}_k) \leq 0 \quad j=1, \cdots m. \qquad (2)$$

Step 3. Updates.

Set $x_{k+1}=x_k + t_k d_k + t_k^2 \tilde{d}_k$.

Compute a new positive definite approximation H_{k+1} of the reduced Hessian matrix using the following modified BFGS formula, due to Powell [10].

$$H_{k+1}=H_k - \frac{H_k S_k S_k^T H_k}{s_k^T H_k s_k} + \frac{z_k z_k^T}{s_k^T z_k}$$

where the quantities involved are defined as follows

$$s_k = x_{k+1} - x_k$$

$$y_k = \nabla_x L(x_{k+1}, \lambda_k, \mu_k) - \nabla_x L(x_k, \lambda_k, \mu_k)$$

$$z_k = \theta y_k + (1-\theta) H_k s_k$$

with

$$\theta = \begin{cases} 1 & \text{if } s_z^T y_k \geq 0.2 s_k^T H_k s_k \\ \dfrac{0.8 s_k^T H_k s_k}{s_k^T H_k s_k - s_k^T y_k} & \text{otherwise.} \end{cases}$$

Set $k = k+1$.

Go back to step 1.

[]

Clearly, algorithm A is largely inspired from the algorithm proposed in [9] for the constrained minimization of a smooth function. The reader is referred to that paper for a motivation of the main ideas. A difference that may not have been intuitively expected is the presence in Algorithm A of constraints on the objectives f_i's in $(\tilde{QP})$, while in [9] this quadratic program only involves the constraint functions. While this set of constraints in $(\tilde{QP})$ is vacuous when only one objective is present (since, in that case, $|I_k| = 1$), its presence is essential in order to eventually achieve a unit step size, a necessary condition for superlinear convergence (see lemma

6 below).

As pointed out earlier, this algorithm is simpler than that obtained by applying to $(\tilde{P})$ the algorithm in [9]. Differences are as follows

(1) Matrices H_k estimate the projection in $\mathbb{R}^n$ of the Hessian of the Lagrangian function, since its other components are trivially known. This idea was already used by Han [7, 8].

(2) The higher order correction in the 'g' constraints in (QP) and $(\tilde{QP})$ involves a vector in $\mathbb{R}^n$, whereas transposition of the algorithm from [9] to problem $(\tilde{P})$ would yield a correction involving a vector in $\mathbb{R}^{n+1}$.

(3) $(\tilde{QP})$ involves a minimization in $\mathbb{R}^n$ rather than $\mathbb{R}^{n+1}$.

(4) The line search test involves f directly instead of the z variable of problem $(\tilde{P})$. It may be of interest to notice, in particular, that after modifications (2) and (3) are performed, a direct transposition of the line search in [9] may not yield a positive step, let alone a unit step.

3. Convergence analysis.

Except for that of lemma 6, proofs are omitted, as they are simple modifications of proofs given in [9].

We suppose that, for any $x \in X$, the sets $\{\nabla f_i(x), i \in I(x)\}$ and $\{\nabla g_j(x), j \in J(x)\}$ are individually made up of linearly independent vectors, where $I(x) \triangleq \{i \text{ s.t. } f_i(x)=f(x)\}$ and $J(x) \triangleq \{j \text{ s.t. } g_j(x)=0\}$, and that there exists some positive constants h_1 and h_2 s.t. $h_1 \leq |H_k| \leq h_2 \forall k \in N$. Under these assumptions, convergence of the algorithm can be proven.

Theorem 1.

Algorithm A either stops at a Kuhn-Tucker point or generates a sequence $\{x_k\}$ for which each accumulation point is a Kuhn-Tucker point for (P).

[]

In order to prove superlinear convergence, we will now further assume that the functions f_i, $i=1, \cdots l$, g_j $j=1, \cdots m$ are three times continuously differentiable. We also suppose that the sequence possesses an accumulation point x^*. In view of theorem 1, the Kuhn-Tucker necessary conditions

$$\nabla_x L(x^*,\lambda^*,\mu^*)=0$$

$$\sum_i \lambda_i^*=1$$

$$\lambda^* \geq 0 \text{ and } \mu^* \geq 0$$

$$g_j(x^*) \leq 0 \quad j=1, \cdots m$$

$$\lambda_i^*(f_i(x^*) - f(x^*))=0 \quad i=1, \cdots l$$

$$\mu_j^* g_j(x^*)=0 \quad j=1, \cdots m$$

are satisfied for some multipliers vectors λ^* and μ^*. Finally, we will suppose that x^* satisfies the second order sufficiency conditions with complementary slackness i.e.

$$\lambda_i^*>0 \text{ iff } f_i(x^*)=f(x^*)$$

$$\mu_j^*>0 \text{ iff } g_j(x^*)=0$$

and $\nabla_{xx} L(x^*,\lambda^*,\mu^*)$ is positive definite on the subspace S^* of directions tangent to the active constraints gradients defined by

$S^* = \{p \text{ s.t. } \exists\ \delta \text{ s.t. } <\nabla f_i(x^*), p> = \delta\ \forall i \in I(x^*) \text{ and } <\nabla g_j(x^*), p> = 0\ \forall\ j \in J(x^*)\}.$

A straightforward result comes from these assumptions.

Lemma 2.

The entire sequence converges to x^*.

[]

In the sequel, we will assume that the sequence of approximate reduced Hessian matrices converges to a matrix whose orthogonal projection on the subspace S^* is equal to the corresponding projection of $\nabla_{xx} L(x^*, \lambda^*, \mu^*)$. The BFGS update used in algorithm **A** is known to generally enjoy such properties.

Lemma 3.

For k large enough,

i) (QP) has a unique solution, $\{d_k^0\} \rightarrow 0$, and $\{d_k\} \rightarrow 0$, where d_k^0 and d_k are computed through step 1 (i) and (ii).

ii) $\{(\lambda_k, \mu_k)\} \rightarrow (\lambda^*, \mu^*)$.

iii) $I_k = I(x^*)$ and $J_k = J(x^*)$.

[]

Lemma 4.

There exist some constants $C_1 > 0$, $C_2 > 0$ and an integer K such that the solutions of (QP_0) and (QP) satisfy

$$C_1 \mid d_k^0 \mid \leq \mid d_k \mid \leq C_2 \mid d_k^0 \mid\ \forall\ k \geq K.$$

[]

Following are two key lemmas leading to the superlinear convergence theorem. Lemma 5 establishes that, in a neighborhood of x^*, the second order method (steps (i), (ii), (iii)), is employed. The result given in lemma 6 relies crucially on the use of the correction $\tilde{d}_k$, resulting in a line search along an arc similar to the line search developed in [11]. It asserts that the Maratos effect [12] is avoided, i.e., the stepsize one is eventually achieved, allowing superlinear convergence to take place. The proof of lemma 6 is not a simple modification of a proof in [9].

Lemma 5.

There exists a positive constant $\bar{\gamma}$ such that, for k large enough, the solution d_k of (QP) satisfies the inequality

$$\theta_k = f(x_k, d_k) - f(x_k) \leq -\bar{\gamma} \mid d_k \mid^2.$$

Thus, in view of lemma 3(i) and lemma 4, the second order direction is always used when x_k is close enough to x^*.

[]

Lemma 6.

For k large enough, the stepsize t_k is one.

Proof.

We will suppose in what follows that k is large enough so that $I(x^*)=I_k$ and $J(x^*)=J_k$. Existence of such a k is guaranteed by lemma 3. For future reference, the Kuhn-Tucker conditions associated with the solution d_k of (QP) are as follows:

$$\sum_{i \in I(x^*)} (\lambda_k)_i \nabla f_i(x_k) + \sum_{j \in J(x^*)} (\mu_k)_j \nabla g_j(x_k) + H_k d_k = 0 \quad (3)$$

$$g_j(x_k) + \langle \nabla g_j(x_k), d_k \rangle \leq - \mid d_k^0 \mid^{\nu} \quad j=1, \cdots m \tag{4}$$

$$(\lambda_k)_i \geq 0, \ (\mu_k)_j \geq 0 \quad i \in I(x^*), \ j \in J(x^*) \tag{5}$$

$$\sum_{i \in I(x^*)} (\lambda_k)_i = 1 \tag{6}$$

As in [11] , it is possible to show that, for k large enough, the correction $\tilde{d}_k$ on the search direction is always well defined and satisfies

$$\tilde{d}_k = O(\mid d_k \mid^2). \tag{7}$$

Two conditions are needed for the line search to yield a unit stepsize, namely feasibility of the resulting point (2) and sufficient decrease (1). Expanding g_j around $x_k + d_k$ and using lemma 3, lemma 4 and (7) gives, for $j \in J(x^*)$,

$$\begin{aligned} g_j(x_k + d_k + \tilde{d}_k) &= g_j(x_k + d_k) + \langle \nabla g_j(x_k + d_k), \tilde{d}_k \rangle + O(\mid d_k \mid^4) \\ &= g_j(x_k + d_k) + \langle \nabla g_j(x_k), \tilde{d}_k \rangle + O(\mid d_k \mid^3) \\ &= - \mid d_k^0 \mid^{\tau} + O(\mid d_k^0 \mid^3). \end{aligned} \tag{8}$$

The last term is negative for k large enough since the sequence $\{d_k^0\}$ converges to zero. Thus the feasibility condition is satisfied.

Expanding f_i around $x_k + d_k$, for $i \in I(x^*)$ and making use of (7), we get

$$f_i(x_k + d_k + \tilde{d}_k) = f_i(x_k + d_k) + \langle \nabla f_i(x_k), \tilde{d}_k \rangle + O(\mid d_k \mid^3).$$

In view of the first set of constraints in $(\tilde{QP})$, this yields

$$f_i(x_k + d_k + \tilde{d}_k) = f_j(x_k + d_k + \tilde{d}_k) + O(\mid d_k \mid^3) \ \forall \ i, j \in I(x^*). \tag{9}$$

Also, since the sequence $\{x_k + d_k + \tilde{d}_k\}$ goes to x^* as k goes to infinity, we have

$$f(x_k + d_k + \tilde{d}_k) = \max_{i \in I(x^*)} \{f_i(x_k + d_k + \tilde{d}_k)\}$$

Thus, using (9) and (6), we obtain, for k large enough,

$$f(x_k+d_k+\tilde{d}_k)=\sum_{i\in I(x^*)}(\lambda_k)_i f_i(x_k+d_k+\tilde{d}_k)+O(|d_k|^3).$$

Expanding the functions f_i around x_k and adding and subtracting $f(x_k)$, we get

$$f(x_k+d_k+\tilde{d}_k)=f(x_k)+\sum_{i\in I(x^*)}(\lambda_k)_i\{f_i(x_k)+<\nabla f_i(x_k),d_k>-f(x_k)+<\nabla f_i(x_k),\tilde{d}_k>$$

$$+\frac{1}{2}d_k^T\nabla_{xx}f_i(x_k)d_k\}+O(|d_k|^3)$$

In view of the definition of θ_k in step 1(ii) of algorithm A, the above can be written

as

$$f(x_k+d_k+\tilde{d}_k)=f(x_k)+\frac{1}{2}\theta_k+\frac{1}{2}\sum_{i\in I(x^*)}(\lambda_k)_i\{f_i(x_k)+<\nabla f_i(x_k),d_k>-f(x_k)\}$$

$$+\sum_{i\in I(x^*)}(\lambda_k)_i\{<\nabla f_i(x_k),\tilde{d}_k>+\frac{1}{2}d_k^T\nabla_{xx}f_i(x_k)d_k\}+O(|d_k|^3).$$

Using (5) and the definition of f, we get

$$f(x_k+d_k+\tilde{d}_k)-f(x_k)\le\frac{1}{2}\theta_k+\frac{1}{2}\sum_{i\in I(x^*)}(\lambda_k)_i<\nabla f_i(x_k),d_k>$$

$$+\sum_{i\in I(x^*)}(\lambda_k)_i\{<\nabla f_i(x_k),\tilde{d}_k>+\frac{1}{2}d_k^T\nabla_{xx}f_i(x_k)d_k\}+O(|d_k|^3).$$

Now, (3) and (7) yield

$$\sum_{i\in I(x^*)}(\lambda_k)_i<\nabla f_i(x_k),d_k>=-\sum_{j\in J(x^*)}(\mu_k)_j<\nabla g_j(x_k),d_k>-d_k^TH_k\,d_k$$

and

$$\sum_{i\in I(x^*)}(\lambda_k)_i<\nabla f_i(x_k),\tilde{d}_k>=-\sum_{j\in J(x^*)}(\mu_k)_j<\nabla g_j(x_k),\tilde{d}_k>+O(|d_k|^3).$$

Plugging these values into the previous expression gives

$$f(x_k+d_k+\tilde{d}_k)-f(x_k)\le\frac{1}{2}\theta_k-\frac{1}{2}\sum_{j\in J(x^*)}(\mu_k)_j<\nabla g_j(x_k),d_k>$$

$$-\sum_{j\in J(x^*)}(\mu_k)_j<\nabla g_j(x_k),\tilde{d}_k>-\frac{1}{2}d_k^TH_k\,d_k+\frac{1}{2}\sum_{i\in I(x^*)}(\lambda_k)_i\,d_k^T\nabla_{xx}f_i(x_k)d_k$$

$$+ O(| d_k |^3) \tag{10}$$

Using lemma 3 and lemma 4, we get

$$f(x_k + d_k + \tilde{d}_k) - f(x_k) \leq \frac{1}{2}\theta_k - \sum_{j \in J(x^*)} (\mu_k)_j < \nabla g_j(x_k), d_k >$$

$$- \sum_{j \in J(x^*)} (\mu_k)_j < \nabla g_j(x_k), \tilde{d}_k > - \frac{1}{2} d_k^T H_k d_k + \frac{1}{2} \sum_{i \in I(x^*)} (\lambda_k)_i d_k^T \nabla_{xx} f_i(x_k) d_k$$

$$- \frac{1}{2} \sum_{j \in J(x^*)} (\mu_k)_j g_j(x_k) + O(| d_k |^\nu) + O(| d_k |^3) \tag{11}$$

Now, since the g_j's are three times continuously differentiable, the relation

$$g_j(x_k + d_k + \tilde{d}_k) = O(| d_k |^\tau) \quad j \in J(x^*)$$

obtained in (8) yields, for $j \in J(x^*)$,

$$g_j(x_k) + < \nabla g_j(x_k), d_k > + < \nabla g_j(x_k), \tilde{d}_k > + \frac{1}{2} d_k^T \nabla_{xx} g_j(x_k) d_k = O(| d_k |^\tau)$$

hence,

$$- \sum_{j \in J(x^*)} (\mu_k)_j < \nabla g_j(x_k), d_k > - \sum_{j \in J(x^*)} (\mu_k)_j < \nabla g_j(x_k), \tilde{d}_k > =$$

$$\sum_{j \in J(x^*)} (\mu_k)_j g_j(x_k) + \frac{1}{2} \sum_{j \in J(x^*)} (\mu_k)_j d_k^T \nabla_{xx} g_j(x_k) d_k + O(| d_k |^\tau).$$

Substituting those values into (11), we obtain

$$f(x_k + d_k + \tilde{d}_k) - f(x_k) - \alpha\theta_k \leq (\frac{1}{2} - \alpha)\theta_k + \frac{1}{2} \sum_{j \in J(x^*)} (\mu_k)_j g_j(x_k)$$

$$+ \frac{1}{2} d_k^T (\sum_{i \in I(x^*)} (\lambda_k)_i \nabla_{xx} f_i(x_k) + \sum_{j \in J(x^*)} (\mu_k)_j \nabla_{xx} g_j(x_k) - H_k) d_k$$

$$+ O(| d_k |^\tau) + O(| d_k |^\nu). \tag{12}$$

Due to the convergence of the projections of the approximate Hessian matrices, we can easily show that the right hand side of (12) is nonpositive for k large enough (see [9] for more details). Thus the 'sufficient decrease' condition is satisfied, and the proof is complete.

Theorem 7.

Under the stated assumptions, the convergence is two-step superlinear, i.e., the following relation holds

$$\lim_{k \to \infty} \frac{| x_{k+2} - x^* |}{| x_k - x^* |} = 0.$$

[]

4. Conclusion.

Many engineering design applications can be transcribed into a formulation that involves the solution of one or several minimax optimization problems. [2] Iterative methods used to solve these problems are required to generate *feasible* iterates that mark an *improvement* of the objective value *at each iteration*. In this paper, we have presented a superlinearly convergent method enjoying these properties.

Whereas algorithm A makes use of a BFGS update formula, our results can be easily extended to the case of other, possibly indefinite, estimates of the Hessian of the Lagrangian. Such an extension is considered in [9] in the case of a single smooth objective function.

Algorithm A is being implemented in the DELIGHT interactive software system for optimization-based design [13, 4]. Preliminary testing on engineering design problems is very promising.

References

[1] Y. Sawaragi, H. Nakayama, and T. Tanino, *Theory of Multiobjective Optimization,* Academic Press (1985).

[2] W.T. Nye and A.L. Tits, "An Application-Oriented, Optimization-Based Methodology for Interactive Design of Engineering Systems," to appear in International Journal of Control (1985).

[3] Z. Ma and A.L. Tits , "Interaction, Specification Refinement, and Tradeoff Exploration in Optimization-Based Design of Engineering Systems," *Proceedings of the 1985 IFAC Workshop on Control Application of Nonlinear Programming and Optimization (Capri, Italy),* Pergamon Press, (1986, to appear).

[4] W.T. Nye, *DELIGHT: An Interactive System for Optimization-Based Engineering Design,* Ph.D. Thesis, Department EECS, University of California, Berkeley, California (1983).

[5] M.K.H. Fan, C.D. Walrath, A.L. Tits, W.T. Nye, M. Rimer, R.T. Grant, and W.S. Levine, "Two Case Studies in Optimization-Based Computer-Aided Design of Control Systems," Presented at presented at the IEEE Control Systems Society 2nd Symposium on Computer-Aided Control System Design (CACSD) , Santa Barbara, California (March 1985).

[6] W.T. Nye, A.Sangiovanni-Vincentelli, J.P. Spoto, and A.L. Tits, "DELIGHT.SPICE: An Optimization-Based System for the Design of Integrated Circuits," *Proceedings of the 1983 Custom Integrated Circuits Conference,* pp. 233-238 (May 1983).

[7] S.P. Han, "Superlinear Convergence of a Minimax Method," TR78-336, Department of Computer Science, Cornell University (1978).

[8] S.P. Han, "Variable Metric Methods for Minimizing a Class of Nondifferentiable Functions," *Math. Prog.* vol. 20, pp. 1-13 (1981).

[9] E.R. Panier and A.L. Tits, "A Superlinearly Convergent Feasible Method for the Solution of Inequality Constrained Optimization Problems," Technical Research Report TR-85-13, Systems Research Center, University of Maryland, College Park, MD 20742 (July 1985).

[10] M.J.D. Powell, "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations," pp. 144-157 in *Numerical Analysis, Dundee, 1977, Lecture Notes in Mathematics 630,* ed. G.A. Watson, Springer-Verlag (1977).

[11] D. Q. Mayne and E. Polak, "A Superlinearly Convergent Algorithm for Constrained Optimization Problems," *Mathematical Programming Study* vol. 16, pp. 45-61 (1982).

[12] N. Maratos, *Exact Penalty Function Algorithms for Finite Dimensional and Optimization Problems,* Ph.D. Thesis, Imperial College of Science and Technology, London, U.K. (1978).

[13] W.T. Nye, E. Polak, A. Sangiovanni-Vincentelli, and A. L. Tits, "DELIGHT: An Optimization-Based Computer-Aided Design System," *Proceedings of the 1981 IEEE International Symposium on Circuits and Systems,* pp. 851-855 (April 1981).