ON DIAGONALLY PRECONDITIONING THE 2-STEPS BFGS METHOD WITH ACCUMULATED STEPS FOR SUPRA-SCALE LINEARLY CONSTRAINED NONLINEAR PROGRAMMING

L. F. ESCUDERO

We present an algorithm for supra-scale linearly constrained nonlinear programming (LCNP) based on the Limited-Storage Quasi-Newton's method. In large-scale programming solving the reduced Newton equation at each iteration can be expensive and may not be justified when far from a local solution; besides, the amount of storage required by the reduced Hessian matrix, and even the computing time for its Quasi-Newton approximation, may be prohibitive. An alternative based on the reduced Truncated-Newton methodology, that has been proved to be satisfactory for super-scale problems, is not recommended for supra-scale problems since it requires an additional gradient evaluation and the solving of two systems of linear equations per each minor iteration. It is recommended a 2-steps BFGS approximation of the inverse of the reduced Hessian matrix such that it does not require to store any matrix since the product matrix--vector is the vector to be approximated; it uses the reduced gradient and solution related to the two previous iterations and the so-termed restart iteration. A diagonal direct BFGS pre--conditioning is used.

1. INTRODUCTION.

Consider the unconstrained nonlinear programming (UNP) problem

$$\text{minimize } \{F(X) \quad X \in R^n\} \tag{1.1}$$

where F(X) is a nonlinear function with the following properties: F(X) is, at least, --2-continuously differentiable; for all $X' \in R^n$ the level sets $L(X') \triangleq \{X : F(X) \leq F(X')\}$ are bounded.

Let $\overset{*}{X}$ denote a weak local minimum in (1.1);- that is, $\overset{*}{X}$ is a point for which $\delta > 0$ such -- that $F(\overset{*}{X}) \leq F(X) \quad X: \; ||X - \overset{*}{X}|| \leq \delta$. (Point $\overset{*}{X}$ will be a strong local minimum if the above relation is a strict inequality for $X \neq \overset{*}{X}$). A first order necessary condition for $\overset{*}{X}$ is that ----- $g(\overset{*}{X}) = 0$, where $g(\overset{*}{X}) \equiv \overset{*}{g}$ is the gradient vector of F(X) evaluated at $\overset{*}{X}$. A sufficient condi--tion for $\overset{*}{X}$ be a weak local minimum is that - $\overset{*}{g} = 0$ and $G(\overset{*}{X})$ is positive definite (pd), where $G(\overset{*}{X}) \equiv \overset{*}{G}$ is the Hessian matrix of F(X) evaluated at $\overset{*}{X}$; note that a second-order necessary condition for $\overset{*}{X}$ is that $\overset{*}{G}$ be a positive semi-definite.

One of the most traditional methods for obtaining $\overset{*}{X}$ is the Newton method; given an initial estimation $X^{(0)}$, it computes a sequence of --- stepdirections $\{d^{(k)}\}$ such that at iteration k it solves the linear system

$$G^{(k-1)} d^{(k)} = -g^{(k-1)} \tag{1.2}$$

and iterates $X^{(k)}$ as follows

$$X^{(k)} = X^{(k-1)} + \alpha^{(k)} d^{(k)} \tag{1.3}$$

where $\alpha^{(k)}$ is the steplength at iteration k, such that F(X) is minimized along direction - $d^{(k)}$,

$$\alpha^{(k)} = \arg \min \{F(X^{(k-1)} + \alpha d^{(k)}) : \quad \alpha > 0\} \tag{1.4}$$

We will refer to (1.3) as the Newton equation and $d^{(k)}$ as the Newton direction.

Problem (1.4) is termed exact linesearch; solving (1.4) is as difficult as solving (1.1); alternatively, the so-termed approximate line search methods are used such that for a direction descent enough, that is

- Laureano F. Escudero - Centro de Investigación UAM-IBM - Pº Castellana, 4 - Madrid-1

- Article rebut el Gener de 1983.

$$(-g^{(k-1)}d^{(k)})/(\|g^{(k-1)}\|_2\|d^{(k)}\|_2) > \varepsilon \quad (1.5)$$

where ε is a positive small tolerance that avoids the quasi-orthogonality between the gradient and the direction, the algorithm is globally convergent if $\alpha^{(k)}$ produces a strong enough reduction in F(X) and, specifically, if the conditions GPW /16/, /24/ are satisfied:

(i) $$|g(X^{(k-1)}+\alpha^{(k)}d^{(k)})^t d^{(k)}| \leq \leq -\eta g(X^{(k-1)})^t d^{(k)} \quad (1.6)$$

or, alternatively /15/, if the calculation of the gradient is expensive,

$$\frac{|F(X^{(k-1)}+\alpha^{(k)}d^{(k)}) - F(X^{(k-1)}+\nu d^{(k)})|}{\alpha^{(k)}-\nu} \leq$$

$$\leq -\eta g(X^{(k-1)})^t d^{(k)} \quad (1.7)$$

where ν is a scalar such that $0\leq\nu<\alpha^{(k)}$, and $0\leq\eta<1$. Note that satisfying (1.6) for $\eta=0$, also satisfies (1.4).

(ii) $$F(X^{(k-1)}) - F(X^{(k-1)}+\alpha^{(k)}d^{(k)}) \geq \geq -\mu\alpha^{(k)} g(X^{(k-1)})^t d^{(k)} \quad (1.8)$$

for $0<\mu\leq 0.5$. Typically, $\eta=0.9$ and $\mu=10E\text{-}04$ such that if $\mu\leq\eta$ $\exists\alpha$ that satisfies conditions GPW. Conditions (1.6) y (1.7) avoid $\alpha^{(k)}$ be excessively small; condition (1.8) avoids it be excessively large.

Newton method is important because if provides a standard with which to compare alternate methods for solving (1.1). Its positive and negative aspects are very well known. Briefly, its advantage is that the algorithm is locally and quadratically convergent; note that a sequence $\{X^{(k)}\}$ is said to converge to $\overset{*}{X}$ with a rate r if

$$0 \leq \lim \frac{\|X^{(k+1)}-\overset{*}{X}\|_r}{\|X^{(k)}-\overset{*}{X}\|_r^r} \equiv \beta < \infty \quad (1.9)$$

The rate of convergence is linear if r=1 (for $0<\beta<1$) and quadratic if r=2. The convergence is Q-superlinear if $\beta=0$ and r=1; the type r for which $\beta>0$ is finite defines the order of this convergence (see e.g. /12/ for most of the concepts used in this work).

The inconveniences of the Newton method are significant; they are as follows:

(i) The method is not globally convergent for $\alpha^{(k)}=1$; that is, $g^{(k)}\neq 0$ for $k\to\infty$ if $X^{(0)}$ is not close enough to $\overset{*}{X}$.

(ii) There is not solution in (1.2) if $G^{(k-1)}$ is singular.

(iii) For convex problems, $G^{(k-1)}$ is not necessarily pd and, then, $d^{(k)}$ is not guaranteed to be descent.

(iv) In any case, the Hessian $G^{(k-1)}$ must be evaluated and a n-dimensional linear system (1.2) must be solved at each iteration k.

In sec. 2 we review the main alternate methods for solving the Newton equation and motivate the 2-steps BFGS method. In sec. 3 we describe the skeletal algorithm for this method. Sec. 4 briefly reviews the LCNP problem. Sec. 5 describes the framework of a given algorithm where the Preconditioned Reduced 2-steps BFGS method with accumulated step is been used. Sec. 6 is devoted to some algorithmic refinements. And, finally, sec. 7 reports some numerical results.

2. MOTIVATION FOR THE 2-STEPS BFGS METHOD WITH ACCUMULATED STEP.

The main alternates to Newton method are as follows.

2.1. QUASI-NEWTON (QN) METHODS /4/.

They are the most reliable methods such that, by calculating the symmetric positive-definite (spd) approximation, say $B^{(k-1)}$ of matrix $G^{(k-1)}$ and using conditions GPW, the Newton difficulties may all be overcome with the exception of (iv); their rate of convergence is Q-superlinear. For non-small problems, their using is prohibitive, at least, with today computer capabilities.

It is well known that the Broyden QN methods and, specifically, the BFGS approximation $B^{(k-1)}$, have been proved to be the QN methods with the best performance. It is also well known that the BFGS approximation $T^{(k-1)}$ of inverse matrix $G^{(k-1)^{-1}}$ is more unstable than the BFGS approximation $B^{(k-1)}$ of $G^{(k-1)}$. (Note that $B^{(k-1)^{-1}}$ is not necessarily $T^{(k-1)}$).

$$T^{(k-1)} = T^{(k-2)} - \frac{p^{(k-1)} y^{(k-1)t} T^{(k-2)} + T^{(k-2)} y^{(k-1)} p^{(k-1)t}}{p^{(k-1)t} y^{(k-1)}} + \left(1 + \frac{y^{(k-1)t} T^{(k-2)} y^{(k-1)}}{p^{(k-1)t} y^{(k-1)}}\right) \frac{p^{(k-1)} p^{(k-1)t}}{p^{(k-1)t} y^{(k-1)}} \quad (2.1)$$

where $p^{(k-1)} \equiv \alpha^{(k-1)} d^{(k-1)} \equiv X^{(k-1)} - X^{(k-2)}$ and $y^{(k-1)} \equiv g^{(k-1)} - g^{(k-2)}$.

$$B^{(k)} = B^{(k-1)} + \frac{y^{(k)} y^{(k)t}}{p^{(k)t} y^{(k)}} + \frac{g^{(k-1)} g^{(k-1)t}}{d^{(k)t} g^{(k-1)}} \quad (2.2)$$

Note that $T^{(k)}$ and $B^{(k)}$ are, at least theoretically, pd provided that $T^{(k-1)}$ and $B^{(k-1)}$ are also pd, respectively and $p^{(k)t} y^{(k)} > 0$; then, $d^{(k)}$ is descent. The BFGS updates are also symmetric pd (spd); satisfy the Quasi-Newton condition; and under mild conditions it can be proved that the path $\{X^{(k)}\} \to \overset{*}{X}$, $k \to \infty$ is globally convergent with a Q-superlinear rate provided that $T^{(0)}$ is spd and conditions GPW are satisfied.

QN methods are used with satisfactory results in constrained nonlinear programming when the number $\bar{n}_S$ of superbasic variables (see /18/ for the definition) is, say $\bar{n}_S \leq 300$. See /5/, /19/, /21/ among others for the nonlinear constrained case and /7/, /18/ for the linearly constrained (LCNP) case.

2.2. CONJUGATE-GRADIENT (CG) METHODS /13/.

This type of methods does not use the Hessian $G^{(k-1)}$, nor its approximation $B^{(k-1)}$. The stepdirection $d^{(k)}$ is obtained by using the negative of the gradient $g^{(k-1)}$ and adding the 'correction' of a multiple of the previous stepdirection. The directions are conjugate provided that F(X) is quadratic and the steplength is obtained with an exact line-search; it has the quadratic termination property, such that the number of iterations to obtain $\overset{*}{X}$ is $m_G \leq n$, where m_G is the number of distinct eigenvalues of Hessian matrix G. It may be preconditioned with an spd matrix, such that the number of distinct eigenvalues of the new matrix is reduced. It is based on the CG method introduced in /17/ for solving systems of linear equations (i.e., the Newton equation (1.2)). The stepdirection can be expressed

$$d^{(r)} = -g^{(r-1)} \quad \text{for } r=1,n+1,\ldots,in+1 \quad (2.3)$$

$$d^{(k)} = -g^{(k-1)} + \beta^{(k)} d^{(k-1)} \quad \text{for } k=2,\ldots,n,n+2,\ldots,in,\ldots \quad (2.4)$$

where

$$\beta^{(k)} = y^{(k-1)t} g^{(k-1)} / y^{(k-1)t} d^{(k-1)} \quad (2.5)$$

If F(X) is a general nonlinear function, a second correction (a multiple of the so-termed restart direction /2/) is usually added to the formula for $d^{(k)}$; in any case, its rate of convergence is quasi linear, but till very recently there were not other methods to cope with supra-scale problems (say, n>600). The stepdirection can be expressed

$$d^{(1)} = -g^{(0)} \quad (2.6)$$

$$d^{(r+1)} = -g^{(r)} + \beta^{(r+1)} d^{(r)} \quad \text{for } r=1,n,2n,\ldots,in \quad (2.7)$$

$$d^{(k)} = -g^{(k-1)} + \beta^{(k)} d^{(k-1)} + \gamma^{(k)} d^{(r)} \quad (2.8)$$

for $k=3,\ldots,n, n+2,\ldots,2n,\ldots,in+2,\ldots,(i+1)n$ where $\beta^{(r+1)}$ and $\beta^{(k)}$ can be expressed by (2.5) and, similarly,

$$\gamma^{(k)} = y^{(r)t} g^{(k-1)} / y^{(r)t} d^{(r)}$$

Conditions GPW do not guarantee $d^{(k)}$ to be descent for $\eta>0$; it is required to add a new condition to be satisfied while obtaining $\alpha^{(k-1)}$ so that additional gradient evaluations are needed /9/, /13/.

CG methods for LCNP are used in /1/, /18/ among others. Due to the restriction to be imposed on $\alpha^{(k-1)}$ such that step $\alpha^{(k-1)} d^{(k-1)}$ is not only descent, but also feasible, it happens very frecuently that conditions GPW and the additional mentioned above are not satisfied; in this case, let $\alpha^{(k-1)}$ be feasible and descent (although, not descent enough) and reset r=k and $d^{(k)} = -g^{(k-1)}$ and hence, the information obtained from previous iterations is lost. See in /9/, /13/ critical revisions of CG-based methods when applied to LCNP problems.

2.3. INEXACT-NEWTON (IN) METHODS /3/.

Since the benefits of the Newton direction are mainly local (i.e., in the vecinity of $\overset{*}{X}$) there appears to be no justification for expending the effort required to get an accurate solution to equation (1.2) when far from

local optimum $\overset{*}{x}$. It makes sense to give an inexact solution $d^{(k)}$ to that equation, increasing the accuracy when getting close to $\overset{*}{x}$, provided that storage and computation are reduced; a scale independent measure of this accuracy is the relative residual error

$$r^{(k)} = \|e^{(k)}\|_2 \; / \; \|g^{(k-1)}\|_2 \qquad (2.10)$$

where $e^{(k)}$ is the residual error vector when solving equation (1.2) with the given accuracy at iteration k, such that it can be written

$$e^{(k)} = B^{(k-1)} d^{(k)} + g^{(k-1)} \qquad (2.11)$$

where $B^{(k-1)}$ is the approximation of $G^{(k-1)}$.

IN methods solve Newton equation (1.2) at each major iteration k, producing a stepdirection $d^{(k)}$ such that condition $r^{(k)} \leq \eta^{(k)}$ is satisfied, where $\eta^{(k)}$ is the given tolerance in the accuracy. They have the following properties:

(i) Global convergence if $\lim \eta^{(k)}=0$ for $k\to\infty$ and $\alpha^{(k)}$ satisfies conditions GPW; the rate of convergence is superlinear depending on $\eta^{(k)}$.

(ii) Stepdirection $d^{(k)}$ is always descent, even when the equation solving is abruptly interrupted.

(iii) Then there is always solution to equation (1.2).

(iv) Hessian matrix $G^{(k-1)}$ alone is not required to be evaluated, nor approximated.

The CG method to solve systems of linear equations introduced in /17/ obtains $d^{(k)}$ by iteratively solving (1.2); at each iteration, say i a stepdirection $d^{(i)}$ is obtained so that $\|e^{(i)}\|_2 < \|e^{(i-1)}\|_2$, where $e^{(i)}$ is defined in (2.11). A so-termed Truncated-Newton (TN) method, a IN method, 'truncates' the sequence of iterations {i} once the given accuracy is obtained. Let 'major iteration' k denote the calculation of steplength $\alpha^{(k)}$ along a given $d^{(k)}$ and the computation of $g^{(k)}$ and $B^{(k)}$. Major iterations are used to modify the algorithm so as to control its global behavior; they play a significant role in early stages of the computation. Let 'minor iteration' i at iteration k denote a given iteration for an iterative solving of Newton or Quasi-Newton equation (1.2) so that stepdirection $d^{(k)}$ is obtained; they are important when close to $\overset{*}{x}$ since, in its vecinity, $\alpha^{(k)}=1$ satisfies conditions GPW and the algorithm possesses the same asymtotic rate of convergence as Newton method. Note that for $\eta^{(k)}$ large enough, the TN method is the same CG method since iteration i=1 satisfies $r^{(i)} \leq \eta^{(k)}$ $\forall k$; for $\eta^{(k)}=0$ the TN method is the same Newton (or QN) method since the equation solving is not truncated. It shares the rate of convergence of the Newton methodology, but it does not require to storage the Hessian matrix $G^{(k-1)}$, nor its approximation $B^{(k-1)}$; it only needs the product $G^{(k-1)}\delta^{(i)}$ (or its approximation), where $\delta^{(i)}$ is the 'minor' stepdirection at iteration i. Therefore, there is a direct tradeoff between the amount of work required to compute a stepdirection and the accuracy with which the Newton equation is solved. This method, 'intermediate' between CG and Newton methodologies, has been used satisfactorily /11/ for solving super-scale problems (say, $300<n\leq 600$ for UNP and $300<\bar{n}_s\leq 600$ for LCNP) with sparse matrices.

The drawback of the TN method when applied to UNP supra-scale problems whose Hessian matrix has not a very special structure, consists in that the additional evaluation that is required for the gradient at each minor iteration i to approximate $G^{(k-1)}\delta^{(i)}$ may be prohibitive. For LCNP prob ems this inconvenience is much stronger since, besides this evaluation (although restricted to the basic-superbasic set of variables, see sec. 5), the solving of two systems of linear equations with the basic matrix is required.

2.4. LIMITED-STORAGE QUASI-NEWTON (LSQN) METHODS /20/.

The BFGS QN stepdirection with exact line search may be interpreted as a CG stepdirection for which the approximation of the Hessian inverse $G^{(k-1)^{-1}}$, instead of being fixed (to the identity matrix in the traditional CG) is updated at each iteration by any member of the Broyden QN methods. This interpretation is of value because it motivates techniques for using limited storage to improve the CG method performance.

Based on the above remark, a new class of methods is proposed, such that the Broyden QN updates (and, specially, the BFGS approximation) of matrix $G^{(k-1)^{-1}}$ use only the last m updates of the previous iterations; for m=1 the m-steps LSQN stepdirection is a CG stepdirection (provided the BFGS formula is used, the previous inverse matrix is the identity and an exact linesearch is used), and for m=k-1 it is a QN stepdirection.

Note that the QN update $T^{(k-1)}$ of $G^{(k-1)^{-1}}$ is not required to be stored, since the step direction $d^{(k)} = -T^{(k-1)}g^{(k-1)}$ (1.2) may be obtained from a sequence of linear combinations of vectors that are computed by using the already available information: point $X^{(j)}$ and gradient $g^{(j)}$ for j=k-m-1,...,k-1, such that

$$T^{(k-m)}g^{(k-1)} = \text{lc}\{T^{(k-m-1)}g^{(k-1)}, T^{(k-m-1)}y^{(k-m)}, p^{(k-m)}\}$$

$$T^{(k-m+1)}g^{(k-1)} = \text{lc}\{T^{(k-m)}g^{(k-1)}, T^{(k-m)}y^{(k-m+1)}, p^{(k-m+1)}\}$$

.....

$$T^{(k-1)}g^{(k-1)} = \text{lc}\{T^{(k-2)}g^{(k-1)}, T^{(k-2)}y^{(k-1)}, p^{(k-1)}\}$$

where $T^{(k-m-1)}=I$; note that $T^{(j-1)}y^{(j)}$ for j=k-1,...,k-m+1 is obtained following the same procedure used for $T^{(j-1)}g^{(k-1)}$. The procedure does not require a great amount of storage for reasonable values of m (say, $2 \le m \le 4$) (see sec. 3), although it could be time consuming; it is more unstable than using $B^{(k-1)}$, but it is prohibitive with Limited Storage.

LSQN methods have some of the properties of QN methods: matrix $T^{(k-1)}$ is spd (if conditions GPW are satisfied), the Quasi-Newton condition is satisfied, and the path $\{X^{(k)}\} \to \overset{*}{X}$, $k \to \infty$ is globally convergent with a superlinear rate. For large values of m these methods have a good rate of superlinear convergence, but the needs for storage and computation are not meaningless; for m=2 there is a good balance between storage and computation to be required and rate of convergence.

A 2-steps LS method has been suggested in /22/ for UNP and used in /23/ for LCNP. It uses information from the last two iterations and the restart iteration. It may be viewed as an extension of the CG method with restart direction that has some of the QN properties. Better results are obtained /13/ with a 2-steps BFGS method, where the initial matrix I is diagonally-BFGS preconditioned and the accumulated step $X^{(k-2)} - X^{(r)}$ is used such that r is the restart iteration.

The advantages of LSQN methods over CG methods are basically, as follows: (a) the step direction is descent provided conditions GPW are satisfied and, then, additional gradient evaluations are not required in the approximate linesearch; and (b) the rate of convergence is much better, even for m=2 where the required storage is still small. Although the convergence is slower than when using QN methods, today there are not other methods to cope with supra-scale problems (say, n>600).

The additional advantage of LSQN methods over CG methods in LCNP is that, since they do not require any additional condition to conditions GPW to guarantee that the superbasic stepdirection, say $d_S^{(k)}$ is descent, it results that it is not to be restarted as $-h^{(k-1)}$ (negative of the reduced gradient, see sec. 4); the reason is that there is more room for a steplength $\alpha^{(k-1)}$ that produces a feasible-descent step, being <u>feasible</u> since it must be $\alpha^{(k-1)} \le \alpha_m^{(k-1)}$ (where $\alpha_m^{(k-1)}$ is the upper bound that guarantees that the bounds on variables and constraints are not violated) and <u>descent enough</u> since it must only satisfy conditions GPW. If there is not such a value, obtain a feasible $\alpha^{(k-1)}$ that be as descent as possible (it will be $\alpha_m^{(k-1)}$ unless for pathological cases), and 'correct' the LSQN matrix if it is not pd; note that conditions GPW guarantee that the matrix is pd so that the stepdirection is descent. Today, there are not other methods to be used for getting practical results when the number $\bar{n}_S$ of superbasic variables at each iteration is, say $\bar{n}_S > 600$ (supra-scale problems).

3. THE DIAGONALLY-PRECONDITIONED 2-STEPS BFGS METHOD WITH ACCUMULATED STEP.

3.1. 2-STEPS BFGS.

Based on sec. 2 it is clear that the LSQN methods (and, specifically, the 2-steps BFGS method) are the most practical approaches

for supra-scale problems without special --- structure. For m=2 and assuming that ------- $T^{(k-3)}=\underline{W}$, where $\underline{W}\equiv W^{-1}$ is a given diagonal matrix (usually, W=I), the computation of --- $d^{(k)} = -T^{(k-1)}g^{(k-1)}$ that saves amount of -- storage and computing time, similarly to /22/ is as follows; see (2.1). (Note W is introduced (see sec. 3.3) to be coherent with /11/).

$$d^{(k)} = -T^{(k-1)}g^{(k-1)}$$

$$= -T^{(k-2)}g^{(k-1)} + \frac{p^{(k-1)t}g^{(k-1)}}{p^{(k-1)t}y^{(k-1)}}T^{(k-2)}y^{(k-1)}$$

$$- \left[\left(1 + \frac{y^{(k-1)t}T^{(k-2)}y^{(k-1)}}{p^{(k-1)t}y^{(k-1)}}\right)\frac{p^{(k-1)t}g^{(k-1)}}{p^{(k-1)t}y^{(k-1)}} - \frac{y^{(k-1)t}T^{(k-2)}g^{(k-1)}}{p^{(k-1)t}y^{(k-1)}}\right]p^{(k-1)} \qquad (3.1)$$

where $T^{(k-2)}g^{(k-1)}$ and $T^{(k-2)}y^{(k-1)}$ can be - written

$$T^{(k-2)}a$$

$$= \underline{W}a - \frac{\bar{\bar{p}}^t a}{\bar{\bar{p}}^t\bar{\bar{y}}}\underline{W}\bar{\bar{y}} + \left[\left(1 + \frac{\bar{\bar{y}}^t\underline{W}\bar{\bar{y}}}{\bar{\bar{p}}^t\bar{\bar{y}}}\right)\frac{\bar{\bar{p}}^t a}{\bar{\bar{p}}^t\bar{\bar{y}}} - \frac{\bar{\bar{y}}^t\underline{W}a}{\bar{\bar{p}}^t\bar{\bar{y}}}\right]\bar{\bar{p}} \qquad (3.2)$$

where $\bar{\bar{p}}\equiv X^{(k-2)}-X^{(k-3)}$, $\bar{\bar{y}}\equiv g^{(k-2)}-g^{(k-3)}$; vector a may represent $g^{(k-1)}$ and $y^{(k-1)}$, alternatively. Assuming that matrix W is diagonal, storage must be provided for $d^{(k)}$, $X^{(j)}$ and $g^{(j)}$ for j=k-1, k-2, k-3, $\underline{W}\bar{\bar{y}}$, $\underline{W}g^{(k-1)}$, ----- $T^{(k-2)}g^{(k-1)}$, $\underline{W}y^{(k-1)}$ and $T^{(k-2)}y^{(k-1)}$. If - intermediate computations are performed as it is described in algorithm A1 (see below), only the intermediate vector UO, the six --- imput vectors (X's and g's), vector W and -- the output vector $d^{(k)}$ are required; but, -- since $\bar{\bar{y}}=g^{(k-2)}-g^{(k-3)}$ will not be required - after step (3) is executed, vector UO can be stored (temporarily) in $g^{(k-3)}$ and, then, -- only seven vectors will be needed besides -- vector W.

SKELETAL ALGORITHM A1

(1) A1:= $\bar{\bar{p}}^t\bar{\bar{y}}$
A2:= $\bar{\bar{y}}^t\underline{W}\bar{\bar{y}}$
A3:= $\bar{\bar{p}}^t y^{(k-1)}$
A4:= $y^{(k-1)t}\underline{W}\bar{\bar{y}}$
A5:= $\bar{\bar{p}}^t g^{(k-1)}$
A6:= $g^{(k-1)t}\underline{W}\bar{\bar{y}}$
A7:= $p^{(k-1)t}g^{(k-1)}$
A8:= $p^{(k-1)t}y^{(k-1)}$

(2) A11:= (1+A2/A1)A3/A1 - A4/A1
A12:= (1+A2/A1)A5/A1 - A6/A1

(3) UO:= $\underline{W}y^{(k-1)}$ - (A3/A1)$\underline{W}\bar{\bar{y}}$ + A11$\bar{\bar{p}}$
$d^{(k)}$:= $\underline{W}g^{(k-1)}$ - (A5/A1)$\underline{W}\bar{\bar{y}}$ + A12$\bar{\bar{p}}$

(4) A9:= $y^{(k-1)t}$UO
A10:= $g^{(k-1)t}$UO

(5) A13:= (1+A9/A8)A7/A8 - A10/A8

(6) $d^{(k)}$:= $-d^{(k)}$ + (A7/A8)UO - A13$p^{(k-1)}$

3.2. ACCUMULATED STEP.

See in /9/, /13/ critical revisions on the - CG method with restart direction; see sec. - 2.2. The major contribution of this methodology could be the expression of the stepdirection $d^{(k)}$ as a linear combination of the gradient (whose weight is -1) of the last -- iteration and the stepdirection of previous iterations such that the weights are $\beta^{(k)}$ for the last iteration, $\gamma^{(k)}$ for the so-termed - restart iteration, and zero for the rest; see (2.8).

It is clear that it is useful to compute a - stepdirection incorporating information from past iterations, provided that the storage - is not strongly increased. It is suggested - /13/ the following procedure; after a sequence (so-termed a cycle) of iterations it ma-- kes sense to consider that a given point has been reached taking only one step along the direction represented by the difference between the point and the solution reached at - the completion of the so-termed restart iteration, say r of the current cycle; a rationale for this hypothesis consists in avoid-- ing the very frequent case of very small sucessive steps $\{\alpha^{(j)}d^{(j)}\}$ when $d^{(j)}$ is obtained by using CG methods. It is assumed that, once point $X^{(k-1)}$ is reached, the stepdirection $d^{(k)}$ will not suffer of the zigzagging phenomenon due to using information from --- iterations k-1 and k-2 (X's and g's), provided that formula (2.4) is substituted by a - 2-steps BFGS-based formula (3.1)-(3.2), ---- where $\bar{\bar{p}}$ and $\bar{\bar{y}}$ accumulate information from -- iterations r,r+1,...,k-2 such that

$$\bar{\bar{p}} = X^{(k-2)} - X^{(r)} = \sum_{j=k-2}^{r+1} p^{(j)} \qquad (3.3)$$

$$\bar{\bar{y}} = g^{(k-2)} - g^{(r)} = \sum_{j=k-2}^{r+1} y^{(j)} \qquad (3.4)$$

Formulation (3.1)-(3.4) will be termed 2BFGSA.

Better results could be obtained by using a m-steps BFGS formula where m=k-1-r; note that the same information is used: $X^{(j)}$ and $g^{(j)}$ for j=r,...,k-1; but it is time consuming. It could be expected that step $X^{(k-2)}-X^{(r)}$ could be a good direction of descent since a substantial reduction in F(X) must have already occurred; otherwise, a new cycle starts with iteration k-1 as the restart iteration.

Let θ_3 be a positive tolerance to determine when a new cycle must begin. At iteration k=0, $\theta_3:=\theta_1$ where θ_1 is a given tolerance. It makes sense to consider that a new cycle starts if the reduction in F(X) during iteration k (i.e., $F^{(k-1)}-F^{(k)}$) is not greater than the fraction θ_3 of the reduction achieved during the current cycle, say c (i.e., $F^{(r+1)}-F^{(k)}$). In that case, the value of θ_3 for the new cycle c+1 will be larger or smaller than the value of the previous one, depending on the reduction in F(X) achieved in the last iteration, say r+1 of the previous cycle, say c-1 (i.e., $F^{(r)}-F^{(r+1)}$) when it is compared with the reduction achieved during the first iteration of the current cycle c (i.e., $F^{(r+1)}-F^{(r+2)}$). If the difference is significantly large (resp. small) then the new value of θ_3 will be smaller (resp.larger) since it is assumed that future reductions will be smaller (resp. larger); otherwise θ_3 is unchanged. The procedure is as follows.

SKELETAL ALGORITHM A2

(0) Data: $X^{(0)}$, $g^{(0)}$, $\theta_3:=\theta_1$, r=k=0.

(1) k:=1
$d^{(1)}:=-g^{(0)}$
Obtain $\alpha^{(1)}$, $X^{(1)}:=X^{(0)}+\alpha^{(1)}d^{(1)}$, $F^{(1)}$ and $g^{(1)}$.

(2) k:=k+1

(3) Obtain $d^{(k)}$.
Use method 1-BFGS, formula (3.1) for k=2.
Use method 2BFGSA for k>2; note that for k=3 it is the usual 2-BFGS.
Obtain $\alpha^{(k)}$, $X^{(k)}$, $F^{(k)}$ and $g^{(k)}$.
If k=2+r, go to (2)

(4) If $F^{(k-1)}-F^{(k)}>\theta_3\,(F^{(r+1)}-F^{(k)})$, go to (2) (3.5)

(5) If $F^{(r)}-F^{(r+1)}>\theta_2\,(F^{(r+1)}-F^{(r+2)})$, $\theta_3:=1/\theta_2$ (3.6)

If $F^{(r)}-F^{(r+1)}\leq 1/\theta_2\,(F^{(r+1)}-F^{(r+2)})$, $\theta_3:=\theta_2$ (3.7)

(6) r:=k-1, go to (2)

Note that $F^{(j)}$ is the value of the objective function on completion of iteration j; when the conditions for the restart procedure are satisfied, the restart iteration r of the new cycle c+1 is the previous iteration (k-1) to the last iteration (k) of cycle c. Typical values /13/, θ_1=0.01 and θ_2=2. Note that algorithm A1 could be used in step (3) of algorithm A2 to obtain $d^{(k)}$ for k>2, provided that $\bar{p}$ and $\bar{y}$ are expressed by (3.3) and (3.4), respectively. In that case, since $g^{(k-2)}$ is not required after the step (3) of algorithm A1 is executed, vector U0 can be stored (temporarily) in $g^{(k-2)}$ and, then, only seven vectors are required besides vector W.

3.3. PRECONDITIONING.

We have stated that the number of iterations to obtain $\overset{*}{X}$ is $m_G\leq n$ if F(X) is quadratic, G is pd, the directions are conjugate and the steplength satisfies (1.4). So, the number of iterations will be reduced if the original problem is substituted by other equivalent problem where $m_R<m_G$, being R the new Hessian matrix.

Let W be an spd matrix. The solution $\bar{d}$ to system (1.2): $\bar{G}d=-\bar{g}$ (and, then, the solution to the related unconstrained quadratic problem) can be obtained by solving system

$$W^{-1/2}\,\bar{G}W^{-1/2}d_W = -W^{-1/2}\,\bar{g} \qquad (3.8)$$

such that $\bar{d}=W^{-1/2}\bar{d}_W$. Let $\bar{R}\equiv W^{-1/2}\,\bar{G}W^{-1/2}$ and, then, $W^{-1/2}\bar{R}W^{1/2}=W^{-1}\bar{G}$. It can be shown /14/ that R has the same eigenvalues as $W^{-1}\bar{G}$. So, a target could be to precondition $\bar{G}$ with a matrix W^{-1}, such that matrix $W^{-1}G$ has a great number of eigenvalues close to unity. Since the condition number κ of a matrix can be expressed as $|\lambda_{max}|/|\lambda_{min}|$, where λ_{max} and λ_{min} are the largest and smallest eigenvalues respectively, it results that smaller κ, greater probability of more (normalized)

eigenvalues close to unity.

Once a suitable matrix W is identified, the CG formula (2.4) could be used to obtain the solution $\bar{d}_W$, where $\bar{g}_W=W^{-1/2}\bar{g}$ and, then, $\bar{y}_W=W^{-1/2}\bar{y}$; such that $\overset{*}{X}=W^{-1/2}\overset{*}{X}_W$. Matrix $W^{-1/2}$ is unstable; it is not required to be used, but $W^{-1}\bar{g}$ whose computation is more stable if W is diagonal or, at least, W is LU-factorized. Matrix W may be identified in several ways /11/; usually, it is updated at each iteration k.

A simple preconditioning that is given good results consists in assigning $W:=\bar{B}^{(k-1)}$, where $\bar{B}^{(k-1)}$ is the diagonal of the BFGS matrix (2.2) $B^{(k-1)}$ such that $\bar{B}^{(0)}=I$ and

$$\bar{B}_\ell^{(k)} = \bar{B}_\ell^{(k-1)} + \frac{1}{y^{(k)t}p^{(k)}}y_\ell^{(k)2} + \frac{1}{g^{(k-1)t}d^{(k)}}g_\ell^{(k-1)2} \qquad (3.9)$$

for $\ell=1,\ldots,n$ and $k\geq 1$. Note that $\bar{B}_\ell$ is the (ℓ,ℓ)-th element of matrix B. The new matrix W requires small storage; it is very stable since:

(a) If $y^{(k-1)t}p^{(k-1)}\leq\varepsilon_2$ or $\exists\bar{B}_\ell^{(k-1)}\leq\varepsilon_2$, $\bar{B}^{(k-1)}$ is not guarantee to be pd; in this case, the matrix is not updated. ε_2 is a small positive tolerance; typically, ε_2=10E-04. Note that the previous update is always pd.

(b) If the condition number κ of $\bar{B}^{(k-1)}$ is large, e.g., $\kappa>\kappa_m$ where $\kappa_m=1/(100n^{1/2}\varepsilon_M)$, ε_M is the machine precision (in our case, 10E-15) and

$$\kappa = \max_{(\ell)}\bar{B}_\ell^{(k-1)}/\min_{(\ell)}\bar{B}_\ell^{(k-1)} \qquad (3.10)$$

$\bar{B}_\ell^{(k-1)}$ is substituted/14/ by $\bar{B}_\ell^{(k-1)^w}$ $\forall\ell$, where $w=\log\kappa_m/\log\kappa$.

4. LINEARLY CONSTRAINED NONLINEAR PROGRAMMING (LCNP)

4.1. INTRODUCTION.

The LCNP problem is

$$\text{minimize}\{F(X)\ X\in F\subset R^n\} \qquad (4.1)$$

where

$$F \triangleq \{X\mid \bar{b}\geq AX\geq\underline{b},\ U\geq X\geq 1\} \qquad (4.2)$$

where A is an m.n matrix, m<n, and F(X) is a general nonlinear twice continuously differentiable function, at least, for feasible points such that for all $\bar{X}\in F$ the level sets

$$L(\bar{X})\triangleq\{X\in F,\ F(X)\leq F(\bar{X})\} \qquad (4.3)$$

are bounded. Let M be the set of constraints, E the set of equality constraints (such that $i\in E$ if $\bar{b}_i=\underline{b}_i$), and J the set of variables. Let $\overset{*}{A}$ be the $\overset{*}{t}$.n matrix of active constraints at a local optimal point, say $\overset{*}{X}$ and $\overset{*}{b}$ the $\overset{*}{t}$-vector of right-hand-side corresponding to $\overset{*}{A}$ (i.e., $\overset{*}{A}\overset{*}{X}=\overset{*}{b}$), such that $\overset{*}{t}=|\overset{*}{W}|$ where $\overset{*}{W}$ is the set of active constraints and $\overset{*}{b}_i=\bar{b}_i\vee\underline{b}_i$ for $i\in\overset{*}{W}$. Let $\overset{*}{V}$ be the set of active variables at $\overset{*}{X}$, such that $j\in\overset{*}{V}$ if $\overset{*}{X}_j=u_j\vee 1_j$ and $\overset{*}{r}=|\overset{*}{V}|$. Let $\overset{*}{I}$ be the $\overset{*}{r}$.n matrix of active bounds at $\overset{*}{X}$, such that it is the n.n identity matrix I from where the row related to variable $j\in\overset{*}{V}$ has been deleted.

Note that $AX-Y=\underline{b}$, $\bar{b}-\underline{b}\geq Y\geq 0\leftrightarrow\bar{b}\geq AX\geq\underline{b}$; then $i\in\overset{*}{W}$ for $\overset{*}{Y}_i=0\vee\bar{b}_i-\underline{b}_i$. The X-variables are termed structural; the Y-variables are termed slack.

Because the constraints are a linear system, the properties of linear subspaces make it possible to state a simple characterization of all feasible moves from a feasible point. Consider the move between two feasible points $\overset{*}{X}$ and $\bar{X}$ along the manifold defined by sets $\overset{*}{W}$ and $\overset{*}{V}$; by linearity $\overset{*}{A}(\overset{*}{X}-\bar{X})=0$ and $\overset{*}{I}(\overset{*}{X}-\bar{X})=0$ since $\overset{*}{A}\overset{*}{X}=\overset{*}{b}$, $\overset{*}{A}\bar{X}=\overset{*}{b}$ and $\overset{*}{X}_j=\bar{X}_j$ $\forall j\in\overset{*}{V}$ and, then,

$$\overset{*}{A}d=0,\ \overset{*}{I}d=0 \qquad (4.4)$$

where d is the stepdirection from $\overset{*}{X}$ to $\bar{X}$ such that $\bar{X}=\overset{*}{X}+\alpha d$. Any vector d for which (4.4) holds is a feasible stepdirection from $\overset{*}{X}$ with respect to the above manifold; it is also termed active stepdirection; it will be descent if $F(\bar{X})<F(\overset{*}{X})$. Steplength α is required to be $0<\alpha\leq\alpha_m$, where α_m defines the maximum allowed steplength such that $\bar{X}$ is still feasible.

Let us define a non-active stepdirection d as the feasible stepdirection such that some constraint or bound is removed from sets $\overset{*}{W}$ and $\overset{*}{V}$, respectively; a feasible stepdirection d is non-active if $\exists i\in M-E\cap\overset{*}{W}$ for which $A_id>0$ if $\overset{*}{Y}_i=0$, $A_id<0$ if $\overset{*}{Y}_i=\bar{b}_i-\underline{b}_i$, or $\exists j\in\overset{*}{V}$ for which $d_j>0$ if $\overset{*}{X}_j=1_j$, $d_j<0$ if $\overset{*}{X}_j=u_j$.

4.2. OPTIMALITY CONDITIONS.

The necessary optimality conditions /10/, /14/

for $\overset{*}{X}$ being a weak local minimum, as stated below, help to assess the LCNP algorithm.

(i) $\overset{*}{X} \epsilon F$ (feasible).

(ii) The reduced gradient vector, say $\overset{*}{h}$ of $F(\overset{*}{X})$ vanishes, such that

$$\overset{*}{h} \equiv \overset{*}{Z}^t \overset{*}{g} = 0 \qquad (4.5)$$

where $\overset{*}{Z}$ is a $n.(n-\overset{*}{t}-\overset{*}{r})$ full column rank matrix, whose columns form the null basis of the range of matrix $(\overset{*}{A}^t, \overset{*}{I}^t)^t$ and, then,

$$\overset{*}{A}\overset{*}{Z} = 0, \quad \overset{*}{I}\overset{*}{Z} = 0 \qquad (4.6)$$

Based on (4.4) and (4.5), we may note - that any linear combination of the co-- lumns of $\overset{*}{Z}$ give an active stepdirection d,

$$d = \overset{*}{Z} d_S \qquad (4.7)$$

where d_S is a $(n-\overset{*}{t}-\overset{*}{r})$-vector termed reduced stepdirection (or superbasic --- stepdirection).

The result (4.5) implies that $\overset{*}{g}$ must be a linear combination of the rows of $\overset{*}{A}$ and $\overset{*}{I}$,

$$\overset{*}{g} = \overset{*}{A}^t \overset{*}{\mu} + \overset{*}{I}^t \overset{*}{\lambda} \qquad (4.8)$$

for some vectors $\overset{*}{\mu}$ and $\overset{*}{\lambda}$; they are termed the Lagrange multipliers of the active constraints and bounds, respectively; viceversa, (4.8) implies (4.5).

(iii) Uniqueness of the Lagrange multipliers.

Let us partition matrix $\overset{*}{A}$ and gradient $\overset{*}{g}$ such that $\overset{*}{A} = (\overset{*}{B}\overset{*}{S}, \overset{*}{N})$ and $\overset{*}{g} = (\overset{*}{g}_{BS}^t, \overset{*}{g}_N^t)^t$, where $\overset{*}{N}$ is a $\overset{*}{t}.\overset{*}{r}$ matrix defined by sets $\overset{*}{W}$ and $\overset{*}{V}$, and $\overset{*}{g}_N$ is the gradient of set $\overset{*}{V}$. Based in (4.8), $\overset{*}{\lambda}$ can be written,

$$\overset{*}{\lambda} = \overset{*}{g}_N - \overset{*}{N}^t \overset{*}{\mu} \qquad (4.9)$$

such that $\overset{*}{\mu}$ satisfies the linear system

$$\overset{*}{g}_{BS} = (\overset{*}{B}\overset{*}{S})^t \overset{*}{\mu} \qquad (4.10)$$

Point $\overset{*}{X}$ does not require A to be a full row rank matrix, but the uniqueness of vectors $\overset{*}{\mu}$ and $\overset{*}{\lambda}$ requires $\overset{*}{B}\overset{*}{S}$ to have -- that property. In any case, computatio nal stability in the algorithm that -- obtain the sequence $\{X^{(k)}\} \to \overset{*}{X}$ requires $\{A_i\}$ to be linearly independent for -- $i \epsilon M$.

(iv) The sign of the Lagrange multipliers - must be as follows.

$\overset{*}{\mu}_i \gtreqless 0$ for $i \epsilon E$ (equality constraint).

$\overset{*}{\mu}_i = 0$ for $i \notin W$ (non-active inequality -- constraint).

$\overset{*}{\mu}_i \geq 0$ for $i \epsilon M-E \cap \overset{*}{W}$ such that $\overset{*}{Y}_i = 0$ (active inequality constraint whose - associated slack variable has the value zero).

$\overset{*}{\mu}_i \leq 0$ for $i \epsilon M-E \cap \overset{*}{W}$ such that $\overset{*}{Y}_i = \bar{b}_i - \underline{b}_i$ (ac tive inequality constraint -- whose associated slack variable takes its upper bound).

$\overset{*}{\lambda}_j = 0$ for $j \notin \overset{*}{V}$ (non-active variable).

$\overset{*}{\lambda}_j \geq 0$ for $j \epsilon \overset{*}{V}$ such that $\overset{*}{X}_j = l_j$ (active va riable at its lower bound).

$\overset{*}{\lambda}_j \leq 0$ for $j \epsilon \overset{*}{V}$ such that $\overset{*}{X}_j = u_j$ (active va riable at its upper bound).

The set $D_1 \cup D_2$, where $D_1 \triangleq \{i \epsilon M-E \cap \overset{*}{W}$ for -- $\overset{*}{\mu}_i = 0\}$ and $D_2 \triangleq \{j \epsilon \overset{*}{V}$ for $\overset{*}{\lambda}_j = 0\}$ is termed - degenerate set of active constraints - and bounds.

(v) Positive semi-definiteness of the ----- Hessian matrix.

The reduced Hessian matrix $\overset{*}{H}$ must be po sitive semi-definite, where

$$\overset{*}{H} \equiv \overset{*}{Z}^t G \overset{*}{Z} \qquad (4.11)$$

If the degenerate set $D_1 \cup D_2$ is not empty then the positive semi-definiteness property of $\overset{*}{G}$ must be extended to the non- active stepdirection d for which it --- holds $A_i d > 0 \wedge \overset{*}{Y}_i = 0 \wedge i \epsilon D_1$, or $A_i d < 0 \wedge \overset{*}{Y}_i = \bar{b}_i - \underline{b}_i \wedge i \epsilon D_1$, or $d_j > 0 \wedge \overset{*}{X}_j = l_j \wedge j \epsilon D_2$, or --- $d_j < 0 \wedge \overset{*}{X}_j = u_j \wedge j \epsilon D_2$.

Conditions (i)-(ii) and (iv)-(v) are necessa ry conditions for local optimality; if Hes-- sian matrix is required in (v) to be positive definite then they are sufficient conditions.

5. SKELETAL ALGORITHM FOR SUPRA-SCALE LCNP PROBLEMS.

Following a traditional approach /18/, let - the active constraints matrix, say $\bar{A}$ be partitioned as

$$\bar{A}d = (\bar{B}, \bar{S}, \bar{N}) \begin{pmatrix} d_B \\ d_S \\ d_N \end{pmatrix} = 0 \qquad (5.1)$$

where the basic stepdirection d_B is used to satisfy the constraints set, the superbasic stepdirection d_S is allowed to vary to mini- mize F(X) (4.1) and the nonbasic stepdirec--

tion d_N is zero, such that set $\bar{V}$ is tempo-- rally fixed at any of their bounds. Here --- $\bar{B}\bar{S} \equiv (\bar{B},\bar{S})$ and $\bar{B}$ is a $\bar{t}.\bar{t}$ nonsingular matrix. At each iteration, the problem then becomes determining vector $d=(d_B^t, d_S^t, d_N^t)^t$ so that it is feasible-descent. Let $\bar{P}$, $\bar{Q}$ and $\bar{V}$ denote - the sets of structural basic, superbasic and nonbasic variables, respectively; and $\bar{P}_s$ -- and $\bar{V}_s$ the sets of slack basic and nonbasic variables, respectively. Let $\bar{s} \equiv |\bar{P}_s|$, $v \equiv |\bar{V}_s|$- and $\bar{n}_s \equiv |\bar{Q}|$ where $\bar{n}_s = n - \bar{t} - \bar{r}$; recall $\bar{t} \equiv |\bar{P}|$ --- and $\bar{r} \equiv |\bar{V}|$. Since $d_N = 0$ and d_S is allowed to be free , it results

$$d_B = -\bar{B}^{-1}\bar{S}d_S \qquad (5.2)$$

such that the variable-reduction characterization of matrix Z can be written

$$Z = \begin{pmatrix} -\bar{B}^{-1}S \\ I \\ 0 \end{pmatrix} \qquad (5.3)$$

so that (4.7) holds.

The unconstrained reduced problem of minimizing F(X) in the manifold defined by $\bar{W}$ and $\bar{V}$ can be expressed as a function of the superbasic set of variables; its quadratic approximation can be written

$$\text{minimize } \{\bar{h}^t d_S + 1/2 d_S^t \bar{H} d_S\} \qquad (5.4)$$

where $\bar{h}$ and $\bar{H}$ are given by (4.5) and (4.11), respectively. Note that $\bar{h}$ can also be written

$$\bar{h} = \bar{g}_S - \bar{S}^t \bar{\mu}_B \qquad (5.5)$$

where $\bar{\mu}_B$ solves the linear system

$$\bar{g}_B = \bar{B}^t \mu_B \qquad (5.6)$$

Note that $\bar{g}_{BS} \equiv (\bar{g}_B^t, \bar{g}_S^t)^t$ where $\bar{g}_B$ and $\bar{g}_S$ are - the basic and superbasic gradients, respectively. Theoretically, the algorithm continues till $\|\bar{h}\| = 0$ or the superbasic set is empty and, then, the de-activating process is executed; the Lagrange multipliers if the solution is 'optimal' or their estimates if the solution is 'quasi-optimal' are used for selecting the set of nonbasic variables to be de-activated.

The active constraints Lagrange multipliers estimates may be obtained by solving system (5.6), basic estimation or by minimizing the residual error on solving system (4.10), basic-superbasic estimation by using the QR -- factorization /6/; see in /8/, /10/ an extensive discussion on the subject. The Lagrange multipliers estimates of the structural nonbasic variables are obtained by using formula (4.9).

While minimizing in a given manifold, it is possible that either a basic or a superbasic variable strikes a bound during the search. If a superbasic variable strikes a bound it becomes nonbasic, the cardinality of the basic-superbasic set (the manifold) is reduced by one, and the search continues. If a basic variable strikes a bound then it is exchanged with an appropiate superbasic variable, and the resulting new superbasic variable is made nonbasic. It is out of the scope of --- this work to describe the different treatment to be given to slack and structural variables, structural unbounded and bounded variables, and structural bounded pure linear, linear with variable coefficient and nonlinear variables. In any case, note that slack variables will only be superbasic while changing the status from basic to nonbasic and - viceversa. A structural variable is defined to be pure linear if its coefficient in the objective function is constant; it is linear with variable-coefficient if, for a given value of the other variables that are used in the same objective function terms, these are a linear function of the given variable; it is nonlinear if, at least, there is an objective function term that is a nonlinear -- function of the given variable if the rest - of the variables are fixed. An example is as follows: $F(X) = 4X_1 + X_2 \log X_3$; variable X_1 is pure linear, variable X_2 is linear with variable-coefficient, and variable X_3 is nonlinear.

SKELETAL ALGORITHM A3.

Let $\bar{H} \equiv H^{(k-1)}$ (it is never computed, not approximated) and $\bar{h} \equiv h^{(k-1)}$. Let $\bar{A} = (\bar{B},\bar{S},\bar{N})$ also include the submatrix related to slack variables (i.e., sets $\bar{P}_s$ and $\bar{V}_s$); $\bar{B}$ is related to sets $\bar{P}$ and $\bar{P}_s$, being an m.m nonsingular matrix; and $\bar{N}$ is related to sets $\bar{V}$ and $\bar{V}_s$, --- being an $m.(\bar{r}+\bar{v})$ matrix.

(0) Data.
$X^{(0)}$, $g^{(0)}$, $F^{(0)}$, $h^{(0)}$, $\theta_1, \theta_2, \theta_3 := \theta_1$, $\bar{B}_S^{(0)} = I$, $k = r = 0$, $\gamma = 0$ (it takes the type of solution

currently obtained, such that $\gamma=1$ if it is 'quasi-optimal', $\gamma=2$ if it is 'optimal', and $\gamma=0$ otherwise, see below), and $k'=0$ (where k' is the current iteration of the subproblem that optimizes the manifold defined by current sets $\bar{V}$ and $\bar{W}$).

Obtain the initial partition (5.1); note that only structural variables are superbasic. Obtain $\bar{\mu}$; since we are dealing with supra-scale problems, only basic estimation $\bar{\mu}_B$(5.6) is used.

If $\bar{Q}=\{\emptyset\}$ go to (10).

The rough procedure for iteration $k\geq 1$ is as follows. See /7/, /11/ for the procedures that are referenced here, but not described in this work; in any case, the algorithm is within the framework proposed in /18/.

(1) LINEAR BASIC-NONBASIC (FEASIBLE-DESCENT) STEP.

Test if set $\bar{P}$ is linear and either there is some linear variable in set $\bar{V}$ or there is some variable in set $\bar{V}_s$ whose associated constraint, say i in set $M-E\cap\bar{W}$ has favorable Lagrange multiplier $\bar{\mu}_i$. If the test is not satisfied, go to (2). Otherwise, analyze if the LP subproblem defined by the basic and linear and slack nonbasic set of variables (by fixing the superbasic and nonlinear nonbasic set to its current value) gives, as its optimal solution, a feasible reduction in F(X). If it is not, go to (2). Otherwise, reset k:=k+1; update solution data; set $\sigma=2$ if a basic-nonbasic exchange has been performed and $\sigma=1$, otherwise; let p be one of the active basic variables; if $p\neq 0$ and $|\bar{Q}|=1$, go to (9); obtain reduced gradient $h^{(k)}$; reset $\bar{B}_S^{(k)}:=I$; if $\sigma=1$ (i.e., the step is feasible-descent without variables exchange), reset $k':=1$; if $\sigma=2$, reset $k':=0$; if $\sigma=1$ and $p=0$, reset $\theta_3:=\theta_1$ and r:=k-1; if $p\neq 0$, go to (9).

(2) STOPPING CRITERIA.

Analyze stopping criteria on optimizing the current reduced unconstrained UNP problem on the given manifold. If they are satisfied then a 'quasi-optimal' or 'optimal' solution to this problem is found, go to (12). The stopping tests (with values true or false) are as follows, provided that the solution is feasible.

t1: $\|\bar{h}\|_2\leq\varepsilon_3\vee\|\bar{d}\|_2/(1+\|\bar{X}+\bar{\alpha}\bar{d}\|_2)\leq\varepsilon_4\vee\bar{Q}=\{\emptyset\}$

t2: $|F(\bar{X})-F(\bar{X}+\bar{\alpha}\bar{d})|/|1+F(\bar{X}+\bar{\alpha}\bar{d})|\leq\varepsilon_5$ in the last τ_3 consecutive iterations.

t3: $\|\bar{h}\|_2\leq\varepsilon_6$

An 'optimal' solution is assumed to be found (and, then, $\gamma=2$) in the current manifold if t1; the current solution is 'quasi-optimal' (and,then, $\gamma=1$) if $\lceil t1\wedge(t2\vee t3)$. Typical values, $\varepsilon_3=\varepsilon_4=\varepsilon_5=10E-04$, $\tau_3=3$, initial $\varepsilon_6=\varepsilon_{10}\|h^{(0)}\|_2$, and $\varepsilon_{10}=0.2$.

(3) STEPDIRECTION.

k:=k+1 and k':=k'+1

Obtain a (descent) superbasic stepdirection $\bar{d}_S\equiv d_S^{(k)}$ on the current UNP and, by using (5.2), obtain a feasible basic stepdirection $\bar{d}_B$. Note that $\bar{d}_S$ is obtained by using problem (5.4) and, then, by 'solving' linear system

$$\bar{H}d_S = -\bar{h}$$

In our case, the method to be used is termed Preconditioned Reduced 2-steps BFGS with accumulated step (PR2SA); see sec. 6.1.

(4) STEPLENGTH.

Obtain the steplength $\bar{\alpha}\equiv\alpha^{(k)}$ such that, by using an approximate linesearch, conditions GPW are satisfied in problem

$$\min\{F(\bar{X}+\alpha\bar{d}):\ 0<\alpha\leq\bar{\alpha}_m\} \qquad (5.8)$$

where $\bar{X}\equiv X^{(k-1)}$ is the feasible solution obtained at the previous iteration, and $\bar{\alpha}_m\equiv\alpha_m^{(k)}$ is the maximum allowed value, at the current iteration, for α such that $\bar{\alpha}\bar{d}$ is still feasible. If there is not any value $0<\alpha\leq\bar{\alpha}_m$ that satisfies conditions GPW, obtain a feasible $\bar{\alpha}$ that be as descent as possible (it will be $\bar{\alpha}_m$, unless for pathological cases).

The approximate linesearch that is been used is the Gill-Murray linesearch (routine GETPTC version 1982) used in /18/; $\eta=0.9$ and $\mu=10E-04$; note that, instead of using $\bar{\alpha}_m$ as a feasible test for the descent enough α, the bound is used within the linesearch.

(5) SOLUTION DATA.

In any case, a new point $X^{(k)}=\bar{X}+\bar{\alpha}\bar{d}$ is obtain

ed and its gradient $(g_B^{(k)t}, g_S^{(k)t})^t$ is evaluated or approximated; $h^{(k)}$ and $F^{(k)}$ are also obtained.

(6) PRECONDITIONING MATRIX.

The diagonal BFGS update $\bar{B}_S^{(k)}$ (see sec. 3.3) related to the current reduced UNP is obtained as a function of $\bar{B}_S^{(k-1)}$, $\bar{p}\equiv\bar{\alpha}\bar{d}_S$, $\bar{y}\equiv h^{(k)}-\bar{h}$ and $h^{(k)}$, where $h^{(k)}$ is the reduced gradient (5.5) related to the current iteration; the procedure 6.1 described elsewhere /11/ is used. Reset $\bar{X}:=X^{(k)}$, $\bar{g}:=g^{(k)}$, $\bar{h}:=h^{(k)}$.

If $\bar{\alpha}=\bar{\alpha}_m$, go to (8)

(7) FORCING A NEW CYCLE.

Note $\bar{\alpha}<\bar{\alpha}_m$, i.e., there is not any active (basic or superbasic variable). For k'>0, analyze if a restart iteration mus be forced; update θ_3, r and k' if needed. See sec. 6.2. In any case, go to (2).

(8) Identify the active (structural or slack) variable, say p. LINEAR BASIC-SUPERBASIC (FEASIBLE-DESCENT) STEP.

If p is a nonlinear variable, test if set $\bar{P}\cup\bar{Q}/\{p\}$ is linear; if it is not, go to (9). Otherwise, analyze if the LP subproblem defined by linear and slack set of variables (by fixing the nonlinear (nonbasic) set at its current value) gives, as its optimal solution, a feasible reduction in F(X). If it is found, obtain the active basic variable p, if any, reset k':=0, k:=k+1, and go to (10) (note that now $Q=\{\emptyset\}$).

(9) ACTIVE BASIC-SUPERBASIC VARIABLE. STATUS CHANGE.

If $p\in\bar{P}\cup\bar{P}_s$ and $|\bar{Q}|=1$, execute an special procedure for pivoting and variables exchange; note that after the pivoting and status updating have been performed, $\bar{Q}=\{0\}$. Go to (10).

If $p\in\bar{P}\cup\bar{P}_s$ (i.e., it is basic) obtain the related row in matrix $\bar{B}^{-1}\bar{S}$ and select the superbasic (structural) variable, say q more suitable (where linear variables have some priority over nonlinear variables) for performing the pivoting (p,q); update diagonal matrix $\bar{B}_S\equiv\bar{B}_S^{(k)}$ with the procedures 6.2 and 6.5 described in /11/; perform the pivoting (p,q); update Lagrange multipliers vector $\bar{u}$ and reduced gradients $h^{(k)}$, $h^{(k-1)}$ and $h^{(r)}$; update sets $\bar{P}_s$, $\bar{P}$ and $\bar{Q}$ such that $\bar{P}_s\leftarrow\bar{P}_s/\{p\}$ if p is slack, $\bar{P}\leftarrow\bar{P}/\{p\}$ if p is structural -- and, in any case, $\bar{P}\leftarrow\bar{P}\cup\{q\}$ and $\bar{Q}\leftarrow\bar{Q}\cup\{p\}/\{q\}$; update matrices $\bar{B}$ (note that it is LU-factorized) and $\bar{S}$. Note that $h^{(k-1)}$ and $h^{(r)}$ are not updated for k'=0, nor $h^{(r)}$ for k'=1, since they will not be required.

If $p\in\bar{Q}$ (i.e., active variable is superbasic) reduce matrix $\bar{B}_S^{(k)}$ and gradients $h^{(k)}$, $h^{(k-1)}$ and $h^{(r)}$; update sets $\bar{Q}$, $\bar{V}$ and $\bar{V}_s$ such that $\bar{Q}\leftarrow\bar{Q}/\{p\}$, $\bar{V}\leftarrow\bar{V}\cup\{p\}$ if p is structural and $\bar{V}_s\leftarrow\bar{V}_s\cup\{p\}$ if p is slack; and rearrange matrix $\bar{A}$.

If there are more active variables, select one of them and go to (8).

(10) SUPERBASIC SET EMPTY.

If $\bar{Q}=\{\emptyset\}$, reset $\gamma=2$ and go to (12).

(11) FORCING A NEW CYCLE.

For k'>0, analyze, as in step (7), if a restart iteration must be forced; update θ_3, k' and r if needed. See sec. 6.2. If the active variable was basic, go to (1); otherwise, go to (2).

(12) DE-ACTIVATING STRATEGY.

Recall that an 'optimal' solution is assumed to be found in the current manifold if t1; the current solution is 'quasi-optimal' if $\lceil t1\wedge(t2\vee t3)$. See step (2).

Let the following tests.

t4: $\|\bar{h}\|_\infty\le\varepsilon_7|\lambda_j|$ $j\in\bar{V}\cup\bar{V}_s$

t5: $|\bar{\lambda}_j|>\varepsilon_8$ $j\in\bar{V}\cup\bar{V}_s$

t6: There is not any nonbasic variable with favorable tendency in its zero or near-to-zero Lagrange multiplier estimate, if any.

When the solution on the current manifold is 'quasi optimal', the main features of the anti-zigzagging strategy are as follows. Let D be the set of candidate variables; it is included by the priced non-unsafe variables with favorable Lagrange multipliers estimates such that t4$\wedge$t5. The set of unsafe variables is included by those nonbasic variables that were made basic-superbasic and, again, become nonbasic in the process of ---

obtaining the next 'optimal' solution. For selecting the candidate variables we use the procedures described elsewhere /11/ sec. 16. for multiple and partial pricing if ------- $(\bar{r}+\bar{v})>\tau_5(n+m-|E|)$, where τ_5 is a positive - tolerance; the Lagrange multipliers are partially (exact) normalized /11/; typical values, τ_5=0.1, τ_6=5 and τ_7=0 (multiple pri-- cing factors), τ_8=0.1 and τ_9=30 (partial -- pricing factors), and τ_{10}=0.1 (partial normalization factor).

After the de-activating process (even if -- D=$\{\emptyset\}$), tolerance ε_6 is reset to ---------- $\varepsilon_6:=\min\{\varepsilon_3,\varepsilon_6(1-\varepsilon_9)\}$; the unsafe set is declared empty if $\varepsilon_6=\varepsilon_3$; see step (2).

When the solution on the current manifold - is 'optimal', the unsafe set is declared -- empty and set D is built with the priced -- nonbasic variables with favorable Lagrange multipliers such that t5. If D=$\{\emptyset\}$ and t6 - /10/, stop since the optimal solution of -- the problem is found; if ⌈t6 then the related nonbasic variable is de-activated. Tolerance ε_6 is reset to $\varepsilon_6:=\varepsilon_{10}\|\bar{\lambda}\|_2$, where $\bar{\lambda}$ is the vector of Lagrange multipliers related to set D.

Typical values are ε_7=0.9, ε_8=10E-04 and -- ε_9=0.3.

Note that it is possible |D|>1; it is the - case for which a multiple de-activating strategy is allowed such that as many as possi-- ble nonbasic variables are de-activated up a given bound, say $\min\{\tau_4,\tau_2(\bar{r}+\bar{v})\}$. Although - more computational experience is required, - it seems that allowing |D|>1 has better performance /11/, /23/ than the single de-activating strategy, provided that τ_2 is not too large: typically, τ_2=0.05 and τ_4=20.

(13) NEW SOLUTION AFTER DE-ACTIVATING.

Note that |D|>0. Update k:=k+1. If p≠0 (i.e. $\bar{Q}=\{\emptyset\}$ and there is, at least, one bounding basic variable), select a variable from set D such that its move in the negative direction of its Lagrange multiplier may de-activate the bounding basic variables; go to (4).

If p=0 obtain the new superbasic stepdirection d_S (see sec. 6.3) by adding set D to -- the old superbasic set $\bar{Q}$; update $\bar{A}$ and $\bar{h}$; -- $\bar{B}_S$ is also updated by using the procedures - 6.4 and 6.6 described elsewhere /11/; obtain the new basic stepdirection d_B, steplength α and the new solution data; reset k':=0 (a -- stepeest descent superbasic stepdirection - will be forced). If there is any slack variable just de-activated, change its status so that it becomes basic and, then, set $\bar{P}_S$ is - increased and a similar procedure to step -- (9) is executed. In any case, go to (8).

6. ALGORITHM REFINEMENTS.

6.1. PRECONDITIONED REDUCED 2-STEPS BFGS WITH ACCUMULATED STEP (PR2SA)

Step (3) of algorithm A3 uses method PR2SA - for obtaining the superbasic stepdirection.- For k'=1 a stepeest descent superbasic stepdirection is forced and, then, $\bar{d}_S:=-\bar{h}$.

For k'=2 formula 1-BFGS (3.1) is used, where $T^{(k-2)}$ is assumed to be $\underline{W}\equiv\bar{B}_S^{(k-1)^{-1}}$, $g^{(k-1)}$ - is substituted by $h^{(k-1)}$, $y^{(k-1)}=h^{(k-1)}-h^{(k-2)}$ and $p^{(k-1)}=\alpha^{(k-1)}d_S^{(k-1)}$. In a similar way to algorithm A1, and using its same notation, -- the procedure is as follows.

SKELETAL ALGORITHM A4

(1) A7:= $p^{(k-1)^t}h^{(k-1)}$
    A8:= $p^{(k-1)^t}y^{(k-1)}$
(3) U0:= $\underline{W}y^{(k-1)}$
    $d_S^{(k)}$:= $\underline{W}h^{(k-1)}$
(4) A9:= $y^{(k-1)^t}$U0
    A10:= $h^{(k-1)^t}$U0
(5) A13:= (1+A9/A8)A7/A8 - A10/A8
(6) $d_S^{(k)}$:= $-d_S^{(k)}$ + (A7/A8)U0 - A13$p^{(k-1)}$

For k'>2 method 2BFGSA is used (see sec. 3.2) and, then, algorithm A1 is applied, where --- $\underline{W}\equiv\bar{B}_S^{(k-1)^{-1}}$, g is substituted by h, and $\bar{\bar{p}}$ and $\bar{\bar{y}}$ can be expressed as follows; see (3.3) and (3.4).

$$\bar{\bar{p}} = X_S^{(k-2)}-X_S^{(r)} \qquad (6.1)$$

where $X_S^{(j)}$ is the superbasic solution obtained at iteration j.

$$\bar{\bar{y}} = h^{(k-2)}-h^{(r)} \qquad (6.2)$$

6.2. FORCING A NEW CYCLE FOR k'>0.

At a given iteration k, steps (7) and (11) of

algorithm A3 test if a new cycle must begin. If k'=1 (i.e., a new cycle has been previously forced), reset $\theta_3:=\theta_1$. For k'>2, test (3.5) is performed; if it is not satisfied, reset k':=1 (i.e., a new cycle is been forced) and use (3.6)-(3.7) to update restart tolerance θ_3 if needed. In any case, reset r:=k-1 for k'=1; recall that r is the restart iteration.

6.3. 'SOLVING' NEWTON EQUATION AFTER DE-ACTIVATING.

Step (13) of algorithm A3 uses the following strategy for obtaining the new $d_S=(\underline{d}_S^t,\underline{\underline{d}}_S^t)^t$ superbasic stepdirection; $\underline{d}_S$ takes the direction related to the old superbasic set $\bar{Q}$ and $\underline{\underline{d}}_S$ is related to set D. The following stepdirection d_S is suggested such that, although the Newton equation $Hd_S=-h$ will not be solved, it is always descent and is not time-consuming.

$$\underline{d}_S = \begin{cases} 0 \text{ if } \bar{Q}=\{\emptyset\} \vee \|\bar{h}\|_2 \le \varepsilon_3 \\ \text{PR2SA direction in } \bar{H}d_S=-\bar{h}, \text{ otherwise} \end{cases} \quad (6.3)$$

$$\underline{\underline{d}}_S = \{-h_q \quad \forall q \in D\} \quad (6.4)$$

where $\underline{d}_S$, $\bar{Q}$ and $\bar{h}$ are related to the old superbasic set (such that $\bar{h}$ is the reduced gradient related to the 'optimal' or 'quasi-optimal' solution $\bar{X}+\bar{\alpha}\bar{d}_S$), $h_q=\bar{\lambda}_q$ and $\varepsilon_3>0$ is a positivo tolerance (it is used in the test t1 of step (2), algorithm A3). Note that d_S (6.3)-(6.4) is a mixture of the stepeest direction and a hopefully accurate Newton direction. Note that the PR2SA direction is obtained by using the algorithms described in sec. 6.1 after updating k':=k'+1.

7. NUMERICAL RESULTS. CONCLUSIONS.

We present here a brief summary of numerical results obtained on three real-life problems; see table 1. Problems I and II involve the management of hydroelectric systems with multiple reservoirs; Problem III is a Lagrangian approach of a scheduling problem of an electrical power system, where the constraints related to the nonlinear transmission losses function have been incorporated to the objective function. Problem I is well-conditioned; Problems II and III are very ill-conditioned. Problem I is typified as a small-scale problem (i.e., $\bar{n}_S \le 300$), Problem II is a super-scale problem (i.e., $300<\bar{n}_S\le 600$) and Problem III is a supra-scale problem (i.e., $\bar{n}_S>600$). Due to the special structure of these problems, the gradient is analytically evaluated.

The problems were tested in the framework of a LCNP algorithm that uses the background of a LP general-purpose package; see sec. 5 and /7/. The methods for obtaining the superbasic stepdirection $\bar{d}_S$ are as follows.

(i) Method RQN. It is based on the update of the R-factor of the BFGS Quasi-Newton approximation (2.2) of the reduced Hessian matrix $\bar{H}=\bar{Z}^t\bar{G}\bar{Z}$ (4.11), so that the Newton equation (1.2) $\bar{R}^t\bar{R}d_S=-\bar{h}$ is solved. It has been designed for small-scale problems /7/.

(ii) Method PRTN. It is based on the diagonally-preconditioning (see sec. 3.3) of the Truncated-Newton method (see sec. 2.3) for solving the Newton equation (5.7) $\bar{H}d_S=-\bar{h}$. It has been designed for super-scale problems /11/.

(iii) Method PR2SA. It is based on the diagonally-preconditioning (see sec. 3.3) of the 2-BFGS Quasi-Newton approximation (see secs. 3.1 and 3.2) of the reduced Hessian matrix $\bar{H}$ (4.11) with accumulated step. It has been designed for supra-scale problems.

See in /11/ the options used for methods RQN and PRTN; see in the above sections the options used for method PR2SA. Tha main differences, aside the proper methods, are summarized as follows:

(a) Method RQN uses both basic based and basic-superbasic based estimations for Lagrange multipliers; the other two methods only use the basic estimation.

(b) Method PR2SA uses multiple and partial pricing and partial (exact) normalization for obtaining the candidate set of nonbasic variables to be de-activated.

Single de-activating strategy was not used. For comparison purposes, the three methods are tested in Problem I; method RQN was not

TABLE 1

PROBLEM DIMENSIONS

Problem	m	|E|	n	Variables partition %			Density %		n_t^*	n_s^*	n_r^*
				Pure L	Non-pure L	Nonlinear	A	G			
I	32	19	88	20	58	22	10	45	26	47	15
II	583	241	1479	24	36	40	1	20	502	459	518
III	702	642	1762	11	25	64	8	40	673	810	279

TABLE 2

RESULTS FOR PROBLEM I

Method	Multiple de-activating strategy	# Major iters.(1)	# Accumulative minor iters.	# Evaluations of F(X), g(X) in major iters.
RQN	up 5 vars.	103	(103)	274
PRTN	up 20 vars.	90	204	263
PR2SA	up 5 vars.	210	(210)	396

TABLE 3

RESULTS FOR PROBLEM II(2)

Method	# Major iters.(1)	# Accumulative minor iters.	# Evaluations of F(X), g(X) in major iters.
PRTN	141	7334	384
PR2SA	4312	(4312)	8431

TABLE 4

RESULTS FOR PROBLEM III(2)

Method	# iters.(1)	# Evaluations of F(X), g(X)
PR2SA	7862	11137

(1) It also takes into account the (LP) iters. required for obtaining the first partition (B, S, N) in step (0) and for solving the other LP subproblems in steps (1) and (8).

(2) Multiple de-activating strategy. Bound: $\text{Min}\{20, 0.05(\bar{v}+\bar{r})\}$

used in Problems II and III; and Problem III was only run with method PR2SA. Tables 2, 3 and 4 report some computational experience on Problem I, II and III, respectively.

The experiment was made in an IBM 370/158 -- with 3MB of real storage and using VM/CMS, - 3MB of virtual storage, the algorithmic ---- tools of MPSX/370 and compiler PL/I OPT(2).

Table 2 shows that when the amount of storage required to save the R-factor of the reduced Hessian matrix is not a great inconvenience, the Quasi-Newton approach is still - the champion.

Problems II and III are very badly scaled -- and, since they are much bigger, we cannot - afford to use the amount of storage required by the reduced Hessian update and, then, method RQN. So, the experiment was only carried out with the other two methods for Pro--- blem II (it is very sparse) and only with method PR2SA for Problem III (matrix A is sparse and matrix G is very dense). Method PRTN is prohibitive in cases as Problem III where at each minor iteration, two m-dimensional - linear systems must be solved (although with the same LU-factorized matrix) and one basic-superbasic gradient evaluation must be performed.

Table 3 shows that for super-scale problems, method PRTN is quite satisfactory; factors L and U of basic matrix B are very sparse and G is sparse too; see in /11/ the procedure - to take advantage of these special structu-- res.

Method PR2SA has the worst performance in -- Problems I and II as it could be expected. - But, for supra-scale problems (as Problem -- III, table 1) where the amount of storage required to save the reduced Hessian (note that $\hat{n}_S^*=810$), the computing time required for gradient evaluations (note that $\hat{n}_S^*+\hat{t}^*=1483$ and - 64% of the structural variables are smooth - nonlinear) and the computing time required - for basic equation solving (note that m=702) are prohibitive, method PR2SA is the only alternative that we have at hand. It requires smaller amount of storage (note that only -- vectors $\bar{B}_S^{(k-1)}$, $X_S^{(j)}$ and $h^{(j)}$ for j=k-1, -- k-2 and r are required for obtaining the superbasic stepdirection $d_S^{(k)}$ at iteration k) and, although the other alternatives would probably require fewer (major) iterations, the computing time and the amount of storage, as far as Problem III is concerned, are still affordable for method PR2SA.

8. REFERENCES.

/1/ R. BENVENISTE, "A quadratic programming algorithm using conjugate directions", Mathematical Programming 16 (1979) 63-80.

/2/ E.M.L. BEALE, "A direction of conjugate gradients" in: F.A. Lootsma (ed.). Numerical methods for nonlinear optimization" (Academic Press., London, 1972) 39-43.

/3/ R.S. DEMBO, S.C. EISENSTAT and T. STEIHAUG, "Inexact Newton methods", SIAM J. of Numerical Analysis 19 (1982) 400-418.

/4/ J.E. DENNIS and J. MORE, "Quasi-Neewton mothods, motivation and theory", SIAM -- Review 19 (1977) 46-89.

/5/ L.F. ESCUDERO, "A projected Lagrangian method for nonlinear programming" IBM -- Scientific Center report G320-3407, Palo Alto (California), 1980.

/6/ L.F. ESCUDERO "An implementation of the QR-Factorization on solving overdetermined systems of linear equations", ----- QUESTIIO 4 (1980) 89-94. See also ----- QR-factorization and its updatings, IBM Scientific Center report SCR-01.82, Ma-- drid, 1982.

/7/ L.F. ESCUDERO, "An algorithm for large-scale quadratic programming and its ex-- tensions to the linearly constrained case, IBM Scientific Center report ------ SCR-01.81, Madrid, 1981

/8/ L.F. ESCUDERO "Lagrange multipliers estimates for constrained minimization, ---- QUESTIIO 5 (1981) 173-186.

/9/ L.F. ESCUDERO "Revisión crítica del método del gradiente conjugado y extensiones en programación nolineal. IBM Scientific Center report PCI-02.82, Madrid,1982.

/10/ L.F. ESCUDERO "Zero or near-to-zero --- Lagrange multipliers in linearly cons-- trained nonlinear programming" QUESTIIO 6 (1982) 193-204.

/11/ L.F. ESCUDERO "On diagonally-precondi-- tioning the Truncated-Newton method for super-scale problems in linearly cons-- trained nonlinear programming". QUESTIIO 6 (1982) 261-281.

/12/ R. FLETCHER, "Unconstrained optimiza--- tion" (Wiley, London, 1980).

/13/ P.E. GILL and W. MURRAY, "Conjugate gra dient methods for large-scale nonlinear optimization", Systems Optimization Laboratory, report SOL 79-15, Stanford University, California, 1979.

/14/ P.E. GILL, W. MURRAY and M.H. WRIGHT, - "Practical Optimization" (Academic Press, London, 1981).

/15/ P.E. GILL, W. MURRAY and M.H. WRIGHT,-- "A note on a sufficient decrease for a - non-derivative step-length procedure", Mathematical Programming 23 (1982) 249-352.

/16/ A.A. GOLDSTEIN and J.F. PRICE, "An effec tive algorithm for minimization", Nume-- rische Mathematik 10 (1967) 184-189.

/17/ M.R. HESTENES and E. STIEFEL, "Methods - of conjugate gradients for solving li--- near systems", J. Res. Nat. Bur. Stan--- dards Sec. B 48 (1952) 409-436.

/18/ B. MURTAGH and M. SAUNDERS, "Large-scale linearly constrained optimization", ---- Mathematical Programming 14 (1978) 41-72.

/19/ B. MURTAGH and M. SAUNDERS, "A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints", - Mathematical Programming Study 16 (1982) 84-117.

/20/ J. NOCEDAL, "Updating Quasi-Newton matri ces with limited storage", Mathematics of Computation 35 (1980) 773-782.

/21/ M. J. D. POWELL, "Algorithms for nonli near constraints that use Lagrangian - functions" 14 (1978) 149-160.

/22/ D. F. SHANNO, "Conjugate gradient ---- methods with inexact searches" Mathema tics of Operations Research 3 (1978) - 244-256.

/23/ D. F. SHANNO and R. E. MARSTEN, "Conju gate gradient methods for linearly cons trained nonlinear programming" Mathema- tical Programming Study 16 (1982) 149-161.

/24/ P. WOLFE, "Convergence conditions for - ascent methods," SIAM Review 11 (1969) 226-234.