M2AN. Mathematical modelling and numerical analysis - Modélisation mathématique et analyse numérique

M. A. Gomes-Ruggiero
J. M. Martínez

The column-updating method for solving nonlinear equations in Hilbert space

M2AN. Mathematical modelling and numerical analysis - Modélisation mathématique et analyse numérique, tome 26, nº 2 (1992), p. 309-330

<http://www.numdam.org/item?id=M2AN_1992__26_2_309_0>

MATHEMATICAL MODELLING AND NUMERICAL ANALYSIS
MODÉLISATION MATHÉMATIQUE ET ANALYSE NUMÉRIQUE
(Vol 26, n° 2, 1992, p 309 à 330)

THE COLUMN-UPDATING METHOD FOR SOLVING NONLINEAR EQUATIONS IN HILBERT SPACE (*)

M. A. GOMES-RUGGIERO [1], J. M. MARTÍNEZ [1]

Communicated by F ROBERT

Abstract — *In 1984, Martinez introduced the Column-Updating method for solving systems of nonlinear equations In this paper we formulate this method for the solution of nonlinear operator equations in Hilbert spaces We prove a local superlinear convergence result We describe a new implementation for large-scale sparse finite dimensional problems and we present a numerical comparison of this implementation against Broyden's method and Schubert's method*

Key Words Quasi-Newton methods , sparse problems , nonlinear equations

Résumé — *La méthode « Column-Updating » pour la résolution des équations non linéaires dans les espaces de Hilbert*
En 1984, Martínez a introduit la méthode « column-updating » pour la résolution des équations non linéaires Dans cet article, nous formulons cette méthode pour la résolution de ces équations dans des espaces de Hilbert Un résultat de convergence superlinéaire est démontré Nous présentons une nouvelle mise en œuvre pour de grands problèmes à matrices éparses en dimension finie et aussi une comparaison numérique entre cette méthode et celles de Broyden et Schubert

1. INTRODUCTION

In 1984, Martinez [17] introduced the Column-Updating method (CUM) for solving systems of nonlinear equations. (See [8, 21, 26]). CUM is a quasi-Newton method where, at each iteration, the column of the Jacobian

(*) Received April 1990
AMS 65H10
[1] Applied Mathematics Laboratory, IMECC-UNICAMP, Caixa Postal 6065, 13 081 - Campinas, SP, Brazil

approximation corresponding to the largest coordinate of the previous increment is replaced in order to satisfy the classical secant equation (see [7]). Martínez presented in [17] some promising numerical experiments for small-dimensional problems.

In the last few years, we have been using the Column-Updating Method for solving practical problems which involve large-scale nonlinear systems of equations.

We obtained very good numerical results, in comparison to other algorithms designed to solve the same type of problems.

These results seemed to be surprising, since the local convergence theory available for CUM imposes that the secant approximation B_k must be the true derivative $F'(x^k)$ when k is multiple of a fixed integer q. This condition is not necessary for many methods for solving nonlinear simultaneous equations. (See [2, 3, 4, 6, 7, 9, 18, 19]).

In order to understand the behavior of CUM for very large finite dimensional problems, we decided to investigate its properties in the infinite dimensional case. Such an investigation should give some insight into the behavior of the finite-dimensional algorithm for discretized infinite-dimensional problems, if discretizations are rather fine.

The behavior of Broyden's method [2], which is the most popular quasi-Newton method for nonlinear systems, in Hilbert spaces was studied in [5, 13, 22, 23]. Under suitable hypothesis, the following results are obtained :

a) Local linear convergence : if the initial point x^0 is close enough to the solution x^k, the sequence of approximations (x^k) converges to the solution with a linear rate.

b) Weak superlinear convergence : for all v belonging to the domain space,

$$\lim_{k \to \infty} \frac{\langle v, x^k - x^* \rangle}{\|x^k - x^*\|} = 0 .$$

c) Strong superlinear convergence : if the difference of the initial derivative approximation and the derivative at x^* is a Hilbert-Schmidt operator (see [15]), we have

$$\lim_{k \to \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} = 0 .$$

The first result (linear convergence) depends strongly on a Bounded Deterioration Principle (see [4, 7]) which may be formulated in the space of linear bounded operators using the natural norm of this space. We observe that in Schubert's method, and other finite-dimensional methods, bounded deterioration principles are formulated in terms of the Frobenius norm (see [18, 19]), whose natural generalization to Hilbert spaces is the norm of Hilbert-Schmidt. Therefore, we do not know if *a*)-*b*)-*c*) hold for the sparse Broyden (Schubert) method, which is also a very popular algorithm for nonlinear equations.

The restrictive hypothesis *c*) for superlinear convergence of Broyden's method encouraged us to extend the finite-dimensional theory of CUM to the infinite dimensional case. In fact, the class of Hilbert-Schmidt operators is a very small class and so, the hypothesis on the initial error operator seems to be very restrictive. Therefore, it seems to be natural to obtain strong superlinear convergence results for Broyden's method through a modification which imposes that a restart must be performed, say, every q iterations.

Moreover, this « restart restriction » for Broyden's method is necessary in practical implementations for large-scale finite dimensional problems. In fact, sparsity of the Jacobian matrix is not preserved by Broyden approximations B_k. Therefore, good implementations of Broyden's method (see [8, 20]) don't store the current Jacobian approximation, but the vectors which define the successive rank-one corrections to this approximation. Hence, both storage and computer time increase at each iteration and the necessity of restarts follows from this fact. Storage and computing-time economy is also obtained using a strategy of dropping old updates, but higher speed of convergence is achieved using Newton restarted iterations.

This paper is organized as follows :

In Section 2 we define the infinite dimensional version of CUM. In Section 3, we prove local strong superlinear convergence of the algorithm defined in Section 2. In Section 4 we introduce a new implementation of CUM for large-scale nonlinear problems. In Section 5 we present a numerical comparison of this implementation of CUM, against Broyden's method and Schubert's method. Some conclusions are drawn in Section 6.

2. STATEMENT OF THE ALGORITHM

Let X, Y be real Hilbert spaces, $\Omega \subset X$ an open and convex set. Assume that $F : \Omega \to Y$ is such that its Fréchet derivative $F'(x)$ exists for all $x \in \Omega$ (see [14, 15, 16, 21]). We will denote $J(x) = F'(x)$. For given $u \in Y$, $v \in X$ we denote $u \otimes v$ the rank-one operator defined by

$$u \otimes vx = \langle v, x \rangle \, u \quad \text{for all} \quad x \in X . \tag{2.1}$$

Let $\{e_j : j \in \mathbb{N}\}$ be an orthonormal basis of X.

Let q be a positive integer, M a (large) positive number. The Column-Updating method for solving $F(x) = 0$ is defined as follows.

ALGORITHM 2.1 : *Let* $x^0 \in \Omega$ *be an arbitrary initial point. Compute* $B_k = J(x^k)$ *whenever* $k \equiv 0$ (mod. q). *For each* $k = 0, 1, 2, \ldots$, *compute*

$$s_k = -B_k^{-1} F(x^k), \quad x^{k+1} = x^k + s_k \tag{2.2}$$

$$j_k = \text{Argmax} \{ | \langle e_j, s_k \rangle | : j \in \mathbb{N} \}, \tag{2.3}$$

$$\theta_k = \min \left\{ 1, \frac{M | \langle e_{j_k}, s_k \rangle |}{\| s_k \|} \right\}. \tag{2.4}$$

If $k + 1 \not\equiv 0$ (mod. q), *compute*

$$B_{k+1} = B_k + \theta_k \frac{(y_k - B_k s_k) \otimes e_{j_k}}{\langle e_{j_k}, s_k \rangle} \tag{2.5}$$

where

$$y_k = F(x^{k+1}) - F(x^k). \tag{2.6}$$

In Section 3 we will show that, under classical conditions, Algorithm 2.1 is well defined and converges superlinearly to some solution x^*.

Let us finish this section proving that, when B_{k+1} is defined by (2.5), and $\theta_k = 1$, then the classical secant equation $B_{k+1} s = y$ (see [7]) is satisfied.

THEOREM 2.1 : Assume that B_{k+1} is given by (2.5), and $\theta_k = 1$. Then

$$B_{k+1} s_k = y_k. \tag{2.7}$$

Proof : By (2.1)-(2.6), we have

$$\begin{aligned} B_{k+1} s_k &= \left[B_k + \frac{(y_k - B_k s_k) \otimes e_{j_k}}{\langle e_{j_k}, s_k \rangle} \right] s_k \\ &= B_k s_k + \frac{(y_k - B_k s_k) \langle e_{j_k}, s_k \rangle}{\langle e_{j_k}, s_k \rangle} = y_k. \quad \square \end{aligned} \tag{2.8}$$

Remark : Global modifications of Newton-like methods may use a definition for s_k different from (2.2) (see [7]). In fact $s_k = -\lambda_k B_k^{-1} F(x^k)$ for some $\lambda_k \neq 0$, if we use steplength strategies, or x^{k+1} lies in a suitable two-dimensional subspace, if we use dogleg-type, or restricted trust-region strategies. The aim of this paper is not to study these possible global modifications. However, let us observe that, if B_{k+1} is chosen according to (2.5) and $\theta_k = 1$, the secant equation (2.7) holds, independently of the definition of s_k. It is easy to see that, if $s_k = -\lambda_k B_k^{-1} F(x^k)$, we have

$$y_k - B_k s_k = F(x^{k+1}) + (\lambda_k - 1) F(x^k). \tag{2.9}$$

3. LOCAL CONVERGENCE RESULTS

We denote $\mathscr{L}(X, Y)$ the space of bounded linear operators $X \to Y$. $\|.\|$ will be the natural norm on this space. Similarly, we denote $\mathscr{L}(Y, X)$ the space of bounded linear operators $Y \to X$.

For proving that the algorithm is well-defined and converges locally to a solution of $F(x) = 0$, we need some additional assumptions on F.

Assumptions on F

Let $x^* \in \Omega$ be such that $F(x^*) = 0$. Assume that :

a) $J(x^*) \in \mathscr{L}(X, Y)$.
b) $J(x^*)^{-1}$ exists and belongs to $\mathscr{L}(Y, X)$.
c) $J : \Omega \to \mathscr{L}(X, Y)$ is continuous.
d) For all $x \in \Omega$, we have

$$\|J(x) - J(x^*)\| \leq L \|x - x^*\| . \tag{3.1}$$

The following lemma is a generalization of Lemma 2.1 of [4].

LEMMA 3.1 : *For all* $x, z \in \Omega$

$$\|F(z) - F(x) - J(x^*)(z - x)\| \leq L \|z - x\| \max \{\|x - x^*\|, \|z - x^*\|\} . \tag{3.2}$$

The main result of this section is the following local convergence theorem.

THEOREM 3.1 : *There exists* $\varepsilon > 0$ *such that, if* $\|x^0 - x^*\| \leq \varepsilon$, *the sequence defined by Algorithm* 2.1 *converges to* x^* *and satisfies*

$$\lim_{k \to \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} = 0 . \tag{3.3}$$

Before proving Theorem 3.1, we need some auxiliary lemmas.

LEMMA 3.2 : *Let* $r_1 \in (0, 1)$. *If* $B \in \mathscr{L}(X, Y)$ *is such that*

$$\|B - J(x^*)\| \leq \frac{r_1}{\|J(x^*)^{-1}\|} \tag{3.4}$$

then, B^{-1} *exists and satisfies*

$$\|B^{-1}\| \leq \frac{\|J(x^*)^{-1}\|}{1 - r_1} . \tag{3.5}$$

Proof : The existence of $B^{-1} \in \mathcal{L}(Y, X)$ follows from classical results in Banach spaces (see, for instance [14, 16]). Now,

$$\begin{aligned}
\|B^{-1}\| &= \|[J(x^*) + (B - J(x^*))]^{-1}\| = \\
&= \|(J(x^*)[I - J(x^*)^{-1}(B - J(x^*))])^{-1}\| \\
&\leq \|J(x^*)^{-1}\| \, \|[I - J(x^*)^{-1}(B - J(x^*))]^{-1}\| \\
&= \|J(x^*)^{-1}\| \left\| \sum_{j=0}^{\infty} [J(x^*)^{-1}(B - J(x^*))]^j \right\| \\
&\leq \|J(x^*)^{-1}\| \sum_{j=0}^{\infty} \|[J(x^*)^{-1}(B - J(x^*))]^j\| \\
&\leq \|J(x^*)^{-1}\| \sum_{j=0}^{\infty} (\|J(x^*)^{-1}\| \, \|B - J(x^*)\|)^j \\
&\leq \|J(x^*)^{-1}\| \sum_{j=0}^{\infty} r_1^j \leq \frac{\|J(x^*)^{-1}\|}{1 - r_1} . \quad \square
\end{aligned}$$

LEMMA 3.3 : *For each $x \in \Omega$, $B \in \mathcal{L}(X, Y)$ let us define*

$$\Phi(x, B) = x - B^{-1} F(x) . \tag{3.6}$$

Let $r \in (0, 1)$. Then, there exist ε_1, $\delta_1 > 0$ such that, if $\|x - x^*\| \leq \varepsilon_1$ and $\|B - J(x^*)\| \leq \delta_1$, the function $\Phi(x, B)$ is well-defined and satisfies

$$\|\Phi(x, B) - x^*\| \leq r\|x - x^*\| . \tag{3.7}$$

Proof : Let

$$\delta_1' = \frac{1}{2\|J(x^*)^{-1}\|} .$$

By (3.4)-(3.5), if $\|B - J(x^*)\| \leq \delta_1'$, B^{-1} exists and satisfies

$$\|B^{-1}\| \leq 2\|J(x^*)^{-1}\| . \tag{3.8}$$

Hence, $\Phi(x, B)$ is well-defined if $x \in \Omega$ and $\delta_1 \leq \delta_1'$. Now,

$$\|\Phi(x, B) - x^*\| \leq A_1 + A_2 \tag{3.9}$$

where $\qquad A_1 = \|x - x^* - B^{-1} J(x^*)(x - x^*)\|$

and $\qquad A_2 = \|B^{-1}[F(x) - J(x^*)(x - x^*)]\|$.

Now, by (3.8),

$$\begin{aligned}
A_1 &= \|x - x^* - B^{-1} B(x - x^*) + B^{-1}(B - J(x^*))(x - x^*)\| \\
&\leq \|B^{-1}\| \, \|B - J(x^*)\| \, \|x - x^*\| \\
&\leq 2\|J(x^*)^{-1}\| \, \delta_1 \|x - x^*\| .
\end{aligned} \tag{3.10}$$

Moreover, by (3.8) and the definition of the Fréchet derivative at x^*,

$$A_2 \leqslant \|B^{-1}\| \, \|F(x) - J(x^*)(x - x^*)\| \leqslant 2 \|J(x^*)^{-1}\| \, \beta(x) \tag{3.11}$$

where
$$\lim_{x \to x^*} \frac{\beta(x)}{\|x - x^*\|} = 0 .$$

Let ε_1, δ_1 be such that

$$2 \left(\delta_1 + \sup_{\|x - x^*\| \leqslant \varepsilon_1} \left\{ \frac{\beta(x)}{\|x - x^*\|} \right\} \right) \leqslant \frac{r}{\|J(x^*)^{-1}\|} . \tag{3.12}$$

Therefore, by (3.9)-(3.12), we have, for $\|B - J(x^*)\| \leqslant \delta_1$, $\|x - x^*\| \leqslant \varepsilon_1$,

$$\begin{aligned} \|\Phi(x, B) - x^*\| &\leqslant 2 \|J(x^*)^{-1}\| \, \delta_1 \, \|x - x^*\| + 2 \|J(x^*)^{-1}\| \, \beta(x) \\ &= \|J(x^*)^{-1}\| \left(2 \, \delta_1 + \frac{2 \, \beta(x)}{\|x - x^*\|} \right) \|x - x^*\| \\ &\leqslant r \|x - x^*\| . \end{aligned}$$

Hence, the proof is complete. □

Remark : Observe that the Lipschitz condition (3.1) is not used in the proof of Lemma 3.3. The same observation holds for the following Lemmas 3.4 and 3.5, but not for the proof of Theorem 3.1, as we shall see latter.

LEMMA 3.4 : *Assume that* $\|x^0 - x^*\| \leqslant \varepsilon_1$ *and* $\|B_k - J(x^*)\| \leqslant \delta_1$ *for all* $k = 0, 1, 2, \ldots$ *Then,* B_k^{-1} *exists for all* $k \geqslant 0$, *and the sequence defined by*

$$x^{k+1} = x^k - B_k^{-1} F(x^k) \tag{3.13}$$

$k = 0, 1, 2, \ldots$, *converges to* x^* *and satisfies*

$$\|x^{k+1} - x^*\| \leqslant r \|x^k - x^*\| \tag{3.14}$$

for all $k = 0, 1, 2, \ldots$

Proof : Observe that (3.13) implies that $x^{k+1} = \Phi(x^k, B_k)$ for all $k = 0, 1, 2, \ldots$ Then, use an inductive argument and Lemma 3.3. □

LEMMA 3.5 : *In addition to the hypotheses of Lemma* 3.4, *assume that*

$$\lim_{k \to \infty} \|B_k - J(x^*)\| = 0 . \tag{3.15}$$

Then, (x^k) *converges superlinearly to* x^*. *That is,*

$$\lim_{k \to \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} = 0 . \tag{3.16}$$

Proof : Let $r' \in (0, 1)$. By Lemma 3.4 there exist $\varepsilon_1' = \varepsilon_1'(r')$, $\delta_1' = \delta_1'(r')$ such that if $\|x^0 - x^*\| \leq \varepsilon_1'$, $\|B_k - J(x^*)\| \leq \delta_1'$ for all $k \geq 0$, then

$$\frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} \leq r' \tag{3.17}$$

holds for all $k = 0, 1, 2, \dots$

Let k_0' be such that $\|B_k - J(x^*)\| \leq \delta_1'$ for all $k \geq k_0'$. Let k_0'' be such that $\|x^k - x^*\| \leq \varepsilon_1'$ for all $k \geq k_0''$. Let $k_0 = \max \{k_0', k_0''\}$. Define $y^\ell = x^{k_0 + \ell}$, $B_\ell' = B_{k_0 + \ell}$ for $\ell = 0, 1, 2, \dots$ Then, the sequence y^ℓ satisfies the hypothesis of Lemma 3.4. Therefore,

$$\frac{\|y^{\ell+1} - x^*\|}{\|y^\ell - x^*\|} \leq r'$$

for all $\ell \geq 0$.

This means that

$$\frac{\|x^{k_0 + \ell + 1} - x^*\|}{\|x^{k_0 + \ell} - x^*\|} \leq r' \tag{3.18}$$

for all $\ell \geq 0$.

Clearly, (3.18) implies that (3.17) holds for all $k \geq k_0$. Since r' is arbitrary, the latter assertion implies that (3.16) holds. □

The following lemma represents a Bounded Deterioration Principle (see [4, 6, 8]) related to formula (2.5).

LEMMA 3.6 : *Assume that* $x^k, x^{k+1} \in \Omega$ *and that* B_{k+1} *is computed using formula* (2.5). *Then,*

$$\|B_{k+1} - J(x^*)\| \leq (1 + M)\|B_k - J(x^*)\| + ML \max \{\|x^k - x^*\|, \|x^{k+1} - x^*\|\} . \tag{3.19}$$

Proof : By (2.6), we have

$$\begin{aligned} \|B_{k+1} - J(x^*)\| &= \left\| B_k + \frac{\theta_k (y_k - B_k s_k) \otimes e_{j_k}}{\langle e_{j_k}, s_k \rangle} - J(x^*) \right\| \\ &\leq \left\| B_k - J(x^*) + \theta_k \frac{(J(x^*) s_k - B_k s_k) \otimes e_{j_k}}{\langle e_{j_k}, s_k \rangle} \right\| \\ &\quad + \left\| \frac{\theta_k (y_k - J(x^*) s_k) \otimes e_{j_k}}{\langle e_{j_k}, s_k \rangle} \right\| . \end{aligned} \tag{3.20}$$

Now,

$$\left\| B_k - J(x^*) + \frac{\theta_k (J(x^*)\, s_k - B_k\, s_k) \otimes e_{J_k}}{\langle e_{J_k}, s_k \rangle} \right\| =$$

$$= \left\| B_k - J(x^*) - \theta_k \frac{[B_k - J(x^*)]\, s_k \otimes e_{J_k}}{\langle e_{J_k}, s_k \rangle} \right\|$$

$$= \left\| [B_k - J(x^*)] \left[I_X - \theta_k \frac{s_k \otimes e_{J_k}}{\langle e_{J_k}, s_k \rangle} \right] \right\|$$

$$\leqslant \| B_k - J(x^*) \| \left\| I_X - \theta_k \frac{s_k \otimes e_{J_k}}{\langle e_{J_k}, s_k \rangle} \right\| . \tag{3.21}$$

Let us now find a bound for

$$\left\| I_X - \theta_k \frac{s_k \otimes e_{J_k}}{\langle e_{J_k}, s_k \rangle} \right\| .$$

If $z \in X$, we have

$$z = \sum_{j \in \mathbb{N}} c_j\, e_j \quad \text{with} \quad \sum_{j \in \mathbb{N}} c_j^2 = \| z \|^2 .$$

Therefore,

$$\left(I_X - \theta_k \frac{s_k \otimes e_{J_k}}{\langle e_{J_k}, s_k \rangle} \right) z = z - \frac{\theta_k\, c_{Jk}\, s_k}{\langle e_{Jk}, s_k \rangle} . \tag{3.22}$$

Therefore, by (2.4), and (3.22),

$$\left\| \left(I_X - \theta_k \frac{s_k \otimes e_{J_k}}{\langle e_{J_k}, s_k \rangle} \right) z \right\| = \left\| z - \frac{\theta_k\, s_k \langle e_{J_k}, z \rangle}{\langle e_{J_k}, s_k \rangle} \right\|$$

$$\leqslant \| z \| + \frac{\theta_k \| s_k \|}{| \langle e_{J_k}, s_k \rangle |} \| z \| \leqslant \| z \| (1 + M) .$$

Thus,

$$\left\| I_X - \theta_k \frac{s_k \otimes e_{J_k}}{\langle e_{J_k}, s_k \rangle} \right\| \leqslant (1 + M) . \tag{3.23}$$

Now, by (3.2) and (2.4),

$$\left\| \frac{\theta_k (y_k - J(x^*) s_k) \otimes e_{j_k}}{\langle e_{j_k}, s_k \rangle} \right\| \leq \frac{\theta_k}{|\langle e_{j_k}, s_k \rangle|} \|y_k - J(x^*) s_k\|$$
$$\leq \frac{\theta_k \|s_k\|}{|\langle e_{j_k}, s_k \rangle|} L \max \{ \|x^k - x^*\|, \|x^{k+1} - x^*\| \}$$
$$\leq ML \max \{ \|x^k - x^*\|, \|x^{k+1} - x^*\| \} . \quad (3.24)$$

Combining (3.20), (3.21), (3.23) and (3.24) we obtain

$$\|B_{k+1} - J(x^*)\| \leq (1 + M) \|B_k - J(x^*)\| + \\ + ML \max \{ \|x^k - x^*\|, \|x^{k+1} - x^*\| \} ,$$

as we wanted to prove. $\square$

Now, we are able to prove Theorem 3.1.

Proof of Theorem 3.1 : Let $r \in (0, 1)$, and consider ε_1, δ_1 defined in Lemma 3.3. Let us define the functions $\varphi_i(v, t)$, for $v, t > 0$, by the following recursive relations :

$$\varphi_0(v, t) = v$$
$$\varphi_{i+1}(v, t) = (1 + M) \varphi_i(v, t) + MLt \quad (3.25)$$

for $i = 0, 1, \ldots, q - 1$.

Clearly, $\varphi_i(v, t)$ is a continuous function of (v, t) and $\varphi_i(0, 0) = 0$ for all $i = 0, 1, \ldots, q$.

Let $\delta_2 \leq \delta_1$, $\varepsilon_2 \leq \varepsilon_1$ be such that

$$\varphi_i(v, t) \leq \delta_1 \quad (3.26)$$

whenever $0 \leq v \leq \delta_2$, $0 \leq t \leq \varepsilon_2$, for all $i = 0, 1, \ldots, q$.

Finally, define

$$\varepsilon = \min \{ \varepsilon_2, \delta_2/L \} . \quad (3.27)$$

Assume that $\|x^0 - x^*\| \leq \varepsilon$. Let us prove by induction on k that

i) x^{k+1} is well-defined.

ii) $\|x^{k+1} - x^*\| \leq r \|x^k - x^*\|$.

iii) If $k + 1 \equiv j \pmod{q}$, $0 \leq j < q$, then

$$\|B_{k+1} - J(x^*)\| \leq \varphi_j(\|B_{k+1-j} - J(x^*)\|, \|x^{k+1-j} - x^*\|) \\ = \varphi_j(\|J(x^{k+1-j}) - J(x^*)\|, \|x^{k+1-j} - x^*\|) . \quad (3.28)$$

Let us prove i)-ii)-iii) for $k = 0$. We have $B_0 = J(x^0)$, and, by (3.27),

$$\|x^0 - x^*\| \leq \delta_2/L . \tag{3.29}$$

Hence, by (3.1) and (3.29),

$$\|J(x^0) - J(x^*)\| \leq \delta_2 .$$

But $\delta_2 \leq \delta_1$ and $\|x^0 - x^*\| \leq \varepsilon_2 \leq \varepsilon_1$, therefore, by Lemma 2.2, we conclude that x^1 is well-defined and

$$\|x^1 - x^*\| \leq r\|x^0 - x^*\| . \tag{3.30}$$

Now, by Lemma 2.5 and (3.30), we have

$$\begin{aligned}
\|B_1 - J(x^*)\| &\leq \\
&\leq (1 + M)\|B_0 - J(x^*)\| + ML \max \{\|x^0 - x^*\|, \|x^1 - x^*\|\} \\
&\leq (1 + M)\|B_0 - J(x^*)\| + ML\,\|x^0 - x^*\| \\
&= (1 + M)\ \varphi_0(\|B_0 - J(x^*)\|, \|x^0 - x^*\|) + ML\,\|x^0 - x^*\| \\
&= \varphi_1(\|B_0 - J(x^*)\|, \|x^0 - x^*\|) \\
&= \varphi_1(\|J(x^0) - J(x^*)\|, \|x^0 - x^*\|) .
\end{aligned}$$

So, (3.28) is proved for $k = 0$.

Now, let us prove the inductive step. Assume that i)-ii)-iii) are true for all indexes between 0 and $k - 1$.

Assume that $k \equiv j \pmod{q}$, $0 \leq j < q$.

By the inductive hypothesis, we have for all $\ell \leq k$,

$$\|x^\ell - x^*\| \leq \|x^0 - x^*\| \leq \varepsilon = \min \{\varepsilon_2, \delta_2/L\} \leq \varepsilon_1 \tag{3.31}$$

hence,

$$\|B_{k-j} - J(x^*)\| = \|J(x^{k-j}) - J(x^*)\| \leq \delta_2 \leq \delta_1 . \tag{3.32}$$

But, by the inductive hypothesis,

$$\|B_k - J(x^*)\| \leq \varphi_j(\|J(x^{k-j}) - J(x^*)\|, \|x^{k-j} - x^*\|) . \tag{3.33}$$

Therefore, by (3.31), (3.32) and (3.26), we have

$$\|B_k - J(x^*)\| \leq \delta_1 .$$

Thus, since, by (3.31), $\|x^k - x^*\| \leq \varepsilon_2 \leq \varepsilon_1$, we can apply Lemma 3.3 to obtain i) and ii).

Now, if $k+1 \equiv 0 \pmod{q}$, the deduction of (3.28) is trivial.

If $k+1 \equiv j \pmod{q}$, we have, by Lemma 3.6, ii), and (3.33),

$$\begin{aligned}\|B_{k+1} - J(x^*)\| &\leqslant (1+M)\|B_k - J(x^*)\| + ML\,\|x^k - x^*\| \leqslant \\ &\leqslant (1+M)\,\varphi_{j-1}(\|J(x^{k-j+1}) - J(x^*)\|,\ \|x^{k-j+1} - x^*\|) \\ &\quad + ML\,\|x^k - x^*\| \\ &\leqslant (1+M)\,\varphi_{j-1}(\|J(x^{k-j+1}) - J(x^*)\|,\ \|x^{k-j+1} - x^*\|) \\ &\quad + ML\,\|x^{k-j+1} - x^*\| \\ &= \varphi_j(\|J(x^{k+1-j}) - J(x^*)\|,\ \|x^{k+1-j} - x^*\|)\,.\end{aligned}$$

Therefore, i)-ii)-iii) are proved for all $k = 0, 1, 2, \ldots$ Hence, the convergence of x^k to x^* is established. Therefore,

$$\lim_{j\to\infty} \|x^{jq} - x^*\| = 0\,,$$

and, by the continuity of J,

$$\lim_{j\to\infty} \|J(x^{jq}) - x^*\| = 0\,.$$

Hence, by the continuity of φ_i, we have

$$\lim_{j\to\infty} \varphi_i(\|J(x^{jq}) - J(x^*)\|,\ \|x^{jq} - x^*\|) = 0\,.$$

for all $i = 0, 1, \ldots, q$.

Therefore, by (3.28),

$$\lim_{j\to\infty} \|B_k - J(x^*)\| = 0\,,$$

and the superlinear convergence follows applying Lemma 3.5. □

4. IMPLEMENTATION FOR LARGE-SCALE FINITE DIMENSIONAL PROBLEMS

Let us consider in this section the case $X = Y = \mathbb{R}^n$. We use $\|\cdot\| = \|\cdot\|_2$. Assume that $\{e_j,\ j = 1, \ldots, n\}$ is the canonical basis of $\mathbb{R}^n$. Therefore,

$$\sup_j \{|\langle e_j, s_k\rangle|\} = \|s\|_\infty \tag{4.1}$$

and hence,

$$|\langle e_{j_k}, s_k\rangle| \geqslant \frac{1}{\sqrt{n}}\|s_k\| \tag{4.2}$$

for all $k \geqslant 0$.

By (4.1), (4.2), taking $M = \sqrt{n}$, we have

$$\frac{M \left| \langle e_{j_k}, s_k \rangle \right|}{\| s_k \|} \geq 1 \tag{4.3}$$

for all $k \geq 0$, and therefore, $\theta_k = 1$ for all $k \geq 0$. Obviously, B_k and $J(x^k)$ may be interpreted as real $n \times n$ matrices and

$$u \otimes v = uv^T$$

for all $u, v \in \mathbb{R}^n$.

We deduce from (4.3) that the secant equation (2.7) is satisfied for all $k \geq 0$, if $M = \sqrt{n}$. Observe that the matrix B_{k+1} coincides with the matrix B_k except at column j_k. In fact, by (2.5), we see that B_{k+1} is obtained replacing column j_k of B_k by $B_k e_{j_k} + (y_k - B_k s_k)/\langle e_{j_k}, s_k \rangle$.

In [17], Martínez suggested implementing the Column-Updating Method storing the $L - U$ factorization of the matrix B_k and updating this factorization in order to obtain B_{k+1}, using well-known techniques currently used in implementations of Linear Programming algorithms (see [1]). However, this idea has some disadvantages. On one hand the « new column » is not sparse, and therefore the Linear Programming updating schemes can be very time and storage-consuming. On the other hand, if sparsity of $J(x^k)$ is introduced in B_k (setting 0 on the entries of the new column which correspond to null entries of J), the performance of the algorithm deteriorates. This deterioration was observed in practical computations and may be attributed to the fact that, when zeros of $J(x)$ are introduced in B_{k+1}, the secant equation (2.7) no longer holds. Maintaining the secant equation seems to be more important than preserving the true sparsity pattern.

In the present implementation we decided to use a similar approach to the one used by Matthies and Strang in their implementation of Broyden's method. In fact, using the Sherman-Morrison formula (see [12]) we deduce a rank-one modification formula for B_k^{-1}, and we use the new formula for defining an algorithm where n additional storage positions are needed at each iteration, instead of the $2\,n$ additional positions that are necessary in the Matthies-Strang-implementation of Broyden's method. The rank-one modification formula for B_k^{-1} is given in the following lemma.

LEMMA 4.1 : *If*

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)\, e_{j_k}^T}{e_{j_k}^T s_k},$$

then B_{k+1}^{-1} *exists if and only if*

$$e_{j_k}^T B_k^{-1} y_k \neq 0 \tag{4.4}$$

and, in this case,

$$B_{k+1}^{-1} = B_k^{-1} + \frac{(s_k - B_k^{-1} y_k) e_{j_k}^T B_k^{-1}}{e_{j_k}^T B_k^{-1} y_k} . \tag{4.5}$$

Proof : Apply formula (1.1.1) of [12]. □

By (4.5), if the Column-Updating method is used to compute $B_{\ell_q+1}, \ldots, B_k, B_{k+1}$, we obtain the following product-form for B_{k+1}^{-1} :

$$B_{k+1}^{-1} = \left[I + \frac{(s_k - B_k^{-1} y_k) e_{j_k}^T}{e_{j_k}^T B_k^{-1} y_k} \right] \ldots \left[I + \frac{(s_{\ell_q} - B_{\ell_q}^{-1} y_{\ell_q}) e_{j_{\ell_q}}^T}{e_{j_{\ell_q}}^T B_{\ell_q}^{-1} y_{\ell_q}} \right] B_{\ell_q}^{-1} . \tag{4.6}$$

We will use (4.6) to define Algorithm 4.1, which is a finite-dimensional version of Algorithm 2.1.

ALGORITHM 4.1 : *Let* $x^0 \in \Omega$ *be an arbitrary initial point. Given* x^k, *the* k-th *approximation to the solution of the problem, we perform the following steps.*

Step 1 : If $k \equiv 0$ (mod. q), execute steps 2-4. If $k \equiv r$ (mod. q), $1 \leqslant r < q$, go to step 5.

Step 2 : Compute the Jacobian matrix at x^k and set

$$B_k = J(x^k) .$$

Step 3 (Factorization of $J(x^k)$).

Compute L, an unitary lower-triangular matrix, U an upper-triangular matrix, P a permutation, such that

$$PB_k = LU . \tag{4.7}$$

Step 4 (Resolution of linear triangular systems).

Compute $s_k \in \mathbb{R}^n$ solving

$$LUs_k = -PF(x^k) . \tag{4.8}$$

Go to Step 6.

Step 5 (Use (4.6) to complete the computation of s_k (see (4.13) and (4.15))).

Compute

$$s_k = (I + u_{k-1}\, e_{j_{k-1}}^T)\, \tilde{s}_{k-1} \,. \tag{4.9}$$

Step 6 (Normalize the step and compute the new point)

$$\begin{aligned} \bar{s}_k &\leftarrow s_k \\ s_k &\leftarrow \lambda_k\, s_k \end{aligned} \tag{4.10}$$

$$x^{k+1} = x^k + s_k \,. \tag{4.11}$$

(The choice of λ_k will be explained in Section 5).

Step 7 (Computation of $u_k = (s_k - B_k^{-1} y_k)/(e_{j_k}^T B_k^{-1} y_k)$).

Execute Steps 7.1-7.4.

Step 7.1 : Compute

$$j_k = \underset{j}{\text{Argmax}} \left\{ \left| e_j^T s_k \right| \right\} \,. \tag{4.12}$$

Step 7.2 (Computation of $\tilde{s}_k = -B_k^{-1} F(x^{k+1})$).
Execute Steps 7.2.1-7.2.2.

Step 7.2.1 (Computation of $\tilde{s} = -B_0^{-1} F(x^{k+1})$).
Solve $LU\tilde{s} = -PF(x^{k+1})$.
If $k \equiv 0 \pmod{q}$, set $\tilde{s}_k = \tilde{s}$ and go to Step 7.3.

Step 7.2.2 : Assuming that $k \equiv r \pmod{q}$, $1 \leq r < q$, compute

$$\tilde{s}_k = (I + u_{k-1}\, e_{j_{k-1}}^T) \dots (I + u_{k-r}\, e_{j_{k-r}}^T)\, \tilde{s} \,. \tag{4.13}$$

Step 7.3 (Compute $v_k = B_k^{-1} y_k = B_k^{-1} F(x^{k+1}) - B_k^{-1} F(x^k)$).

$$v_k = \bar{s}_k - \tilde{s}_k \,. \tag{4.14}$$

Step 7.4 : Compute

$$u_k = (s_k - v_k)/(e_{j_k}^T v_k) \,. \tag{4.15}$$

Step 8 : $k \leftarrow k + 1$.

Remark : By analyzing one iteration of Algorithm 4.1, we verify that at iteration k of this process, we need :

a) The (sparse) real matrices L and U,
b) The set of (n) indexes which define P,
c) The residual n-vector $F(x^k)$,
d) The n-vectors $u_{k-r}, \dots, u_{k-1}, u_k$,
e) The indexes $j_{k-r}, \dots, j_{k-2}, j_k$.

Observe that, essentially, at each iteration k such that $k \equiv r \pmod{q}$, $1 \leqslant r \leqslant q-1$, we need n additional storage positions in relation to the previous iteration. Similarly, we use $O(n)$ additional flops for computing x^{k+1}.

On the other hand, memory limited implementations of Broyden's method need $2n$ additional real positions per ordinary iteration, and the updating procedure is more expensive that the one described in Algorithm 4.1 (see [20]). Of course, the above observations impose machine dependent limitations on the value of q.

5. NUMERICAL EXPERIMENTS

We wrote FORTRAN codes which implement the Column-Updating Method (CUM), as defined by Algorithm 4.1, Broyden's first method [2] using the idea of Matthies-Strang [20] and Schubert's method (see [3, 19, 24]). All the tests were run in a VAX11/785 at the State University of Campinas, using single precision, the FORTRAN 77 compiler and the VMS Operational System. The implementation of methods for solving sparse nonlinear systems of equations requires a decision about the algorithm which is going to be used for solving the underlying linear systems (for instance, at Step 1.2 of Algorithm 4.1). (See [10]). We used the George-Ng [11] factorization algorithm, which uses a static data structure and a symbolic factorization scheme to predict fill-in in calculations, for all the linear algebra calculations in our codes.

We adopted some safeguards to prevent singularity of matrices B_k:

a) Assume that MACHEPS is the machine precision, SQMAP = (MACHEPS)$^{1/2}$. When computing the $L-U$ factorization of $B_k = (b_{ij}^k)$, if an entry u_{ii} such that $|u_{ii}| \leqslant b = \text{SQMAP} \max_{i,j} \{|b_{ij}^k|\}$ appears, this entry is replaced by $sg(u_{ii})b$. We used the same safeguard in the implementation of the methods of Broyden and Schubert.

b) As is well-known (see [10, 12]) even well-scaled triangular matrices may be very ill-conditioned. Therefore, even after the safeguard *a*), the Newton step may be very large. We prevent our implemented algorithms against large steps providing Δ, an initial estimator of the distance between the initial point and the solution, and computing λ_k in (4.10) in order that $\|x^{k+1} - x^k\| \leqslant \Delta$. Therefore, in (4.10),

$$\lambda_k = \min\left\{1, \frac{\Delta}{\|s_k\|}\right\}. \tag{5.1}$$

The choice (5.1) does not invalidate our convergence theorem since a small enough ε in Theorem 3.1 guarantees that $\|s_k\| \leqslant \Delta$ for all $k \geqslant 0$. Moreover, since $\theta_k \equiv 1$ $(M = \sqrt{n})$ in Algorithm 4.1, the secant equation holds for all $k \geqslant 0$. Globally convergent modifications of Algorithm 4.1 should need more sophisticated choices for λ_k (see [7]). We only intend to compare local versions of the CUM method, Broyden's method and Schubert's method, therefore, we used the same control of steplength (5.1) for the three algorithms.

c) The annihilation of $e_{j_k}^T B_k^{-1} y_k$ in (4.4) corresponds to the annihilation of the j_k-th coordinate of v_k in (4.14). When this happens, the algorithm cannot continue because B_{k+1} is singular. Therefore, after computing v_k in (4.14) we test the inequality

$$\left| e_{j_k}^T v_k \right| \leqslant \mathrm{SQMAP} \, \| v_k \| \, . \tag{5.2}$$

If (5.2) holds, we reset $B_{k+1} = B_k$. We used a similar safeguard in the implementation of Broyden's method.

We used the following stopping criteria :

— Convergence of type 0 ($C0$) : When $\|F(x^k)\|_\infty \leqslant \mathrm{TOL} \; \|F(x^0)\|$.

— Convergence of type 1 ($C1$) : When

$$\|x^{k+1} - x^k\|_\infty \leqslant 10^{-4} \|x^{k+1}\|_\infty + 10^{-25}.$$

— Divergence (D) : When $\|F(x^k)\|_\infty \geqslant 10^4 \|F(x^0)\|$.

— Excess of Iterations (E) : When $k \geqslant 100$.

Let us now describe the test functions used in our comparative study.

Problem 1 (Broyden Tridiagonal [3])

$$\begin{aligned} f_1(x) &= (3 - 2\,x_1)\,x_1 - 2\,x_2 + 1 \\ f_i(x) &= (3 - 2\,x_i)\,x_i - x_{i-1} - 2\,x_{i+1} + 1 \quad i = 2(1)\,n - 1 \\ f_n(x) &= (3 - 2\,x_n)\,x_n - x_{n-1} + 1 \\ x^0 &= (-1, \ldots, -1)^T , \quad \Delta = 10 , \quad \mathrm{TOL} = 10^{-5} . \end{aligned}$$

Problem 2 (Band Broyden [3])

$$f_i(x) = (3 + 5\,x_i^2)\,x_i + 1 + \sum_{j \in I_i} (x_j + x_j^2) , \quad i = 1(1)\,n ,$$

where
$$\begin{aligned} I_i &= \{i_1, \ldots, i_2\} - \{i\} , \\ i_1 &= \max\,\{1, i - 5\} , \quad i_2 = \min\,\{n, i + 5\} \\ x^0 &= (-1, \ldots, -1)^T , \quad \Delta = 10 , \quad \mathrm{TOL} = 10^{-5} . \end{aligned}$$

Problem 3 (Trigexp [27])

$$f_1(x) = 3x_1^3 + 2x_2 - 5 + \sin(x_1 - x_2)\sin(x_1 + x_2)$$

$$f_i(x) = -x_{i-1}e^{(x_{i-1} - x_i)} + x_i(4 + 3x_i^2) + 2x_{i+1} + \sin(x_i - x_{i+1})\sin(x_i + x_{i+1}) - 8\,, \quad i = 2(1)(n-1)$$

$$f_n(x) = -x_{n-1}e^{(x_{n-1} - x_n)} + 4x_n - 3$$

$$x^0 = (0, \dots, 0)^T\,, \quad \Delta = 3\,, \quad \mathrm{TOL} = 10^{-5}\,.$$

Problem 4 (Poisson [25]). This problem is the nonlinear system of equations arising from finite difference discretization of the Poisson boundary problem

$$\begin{aligned} \Delta u &= \frac{u^3}{1 + s^2 + t^2}\,, \quad 0 \leqslant s \leqslant 1\,, \quad 0 \leqslant t \leqslant 1 \\ u(0, t) &= 1 \\ u(1, t) &= 2 - e^{-t}\,, \quad t \in [0, 1] \\ u(s, 0) &= 1 \\ u(s, 1) &= 2 - e^{s}\,, \quad s \in [0, 1]\,. \end{aligned}$$

We used L^2 grids with $L = 15$ and $L = 31$. Therefore $n = 225$ and $n = 961$ respectively

$$x^0 = (-1, \dots, -1)^T\,, \quad \Delta = 5\,, \quad \mathrm{TOL} = 10^{-8}\,.$$

Problem 5 :

$$f_1(x) = -2x_1^2 + 3x_1 - 2x_2 + 0.5\,x_{\alpha_1} + 1$$

$$f_i(x) = -2x_i^2 + 3x_i - x_{i+1} - 2x_{i+1} + 0.5\,x_{\alpha_i} + 1\,, \quad i = 2(1)\,n - 1$$

$$f_n(x) = -2x_n^2 + 3x_n - x_{n-1} + 0.5\,x_{\alpha_n} + 1\,,$$

for α_j, $j = 2(1)\,n$ randomly chosen in $\{\alpha_{j\,\min}, \dots, \alpha_{j\,\max}\}$ where $\alpha_{j\,\min} = \max\{1, j - b\}$, $\alpha_{j\,\max} = \min\{n, j + b\}$ and b is a parameter which defines the bandwith. We used $b = 15$, 30, 50 and 100

$$x^0 = (-1, \dots, -1)^T\,, \quad \Delta = 10\,, \quad \mathrm{TOL} = 10^{-5}\,.$$

We report the results in tables 1 and 2. In this tables STOR means the number of thousands of real positions used by the algorithm, RSTP means the reason for stopping (see Stopping Criteria), ITER is the number of iterations and TIME is the total CPU time (in seconds).

6. CONCLUSIONS

In this paper we presented a new convergence result for the Column-Updating Method for solving nonlinear equations in Hilbert space, a new implementation of this method for large-scale nonlinear systems of equations, and a numerical comparison against Broyden's method and Schubert's method.

The results of the experiments are extremely encouraging. Both in the unrestarted as in the restarted versions of the methods, CUM was clearly the best of the three algorithms. It only looses to Schubert's methods in terms of storage requirements in some situations, but this disadvantage is compensated by its performance in terms of robustness and execution time.

The storage requirements of Broyden's method are always greater than those of CUM. The number of iterations is generally the same for both methods, but CUM wastes less CPU time because a typical iteration of Broyden is more expensive than a typical iteration of CUM. Both Broyden and CUM are more efficient than Schubert in terms of execution time.

TABLE 1

Numerical Comparison of Broyden, Schubert and CUM, without restarts

Problem	n	Broyden				Schubert				CUM			
		STOR	RSTP	ITER	TIME	STOR	RSTP	ITER	TIME	STOR	RSTP	ITER	TIME
1	1 000	33	*C*0	7	1 05	26	*C*1	5	1 64	25	*C*1	6	0 85
1	3 000	99	*C*0	7	3 29	78	*C*1	5	4 86	78	*C*1	6	2 46
1	5 000	165	*C*0	7	5 88	130	*C*1	5	8 20	130	*C*1	6	4 18
1	10 000	330	*C*0	7	12 4	260	*C*1	5	17 6	260	*C*1	6	8 84
1	15 000	495	*C*0	7	21 6	390	*C*1	5	29 0	390	*C*1	6	15 1
1	20 000	660	*C*0	7	30 0	520	*C*1	5	40 3	520	*C*1	6	22 3
2	1 000	71	*C*1	8	2 89	70	*C*1	8	8 54	64	*C*1	8	2 71
2	3 000	213	*C*1	8	8 82	210	*C*1	8	26 1	192	*C*1	8	8 28
2	5 000	355	*C*1	8	17 0	350	*C*1	8	52 0	320	*C*1	8	15 4
2	10 000	710	*C*1	8	42 0	700	*C*1	8	116 0	640	*C*1	8	38 5
3	1 000	133	*C*1	57	25 3	26	*E*	100	37 1	91	*C*1	71	25 0
3	3 000	399	*C*1	57	87 2	78	*E*	100	109 0	273	*C*1	71	77 4
3	5 000	665	*C*1	57	162 0	130	*E*	100	183 0	455	*C*1	71	148 0
4	225	27	*C*1	4	1 05	27	*C*0	4	2 71	26	*C*0	5	1 10
4	961	223	*C*1	4	13 1	222	*C*1	5	51 5	220	*C*1	5	13 9
5 $b =$ 15	1 000	62	*C*1	7	2 50	56	*C*1	6	7 0	56	*C*1	7	2 26
5 $b =$ 30	1 000	91	*C*1	7	4 65	85	*C*1	6	17 4	85	*C*1	7	4 43
5 $b =$ 50	1 000	129	*C*1	7	9 0	123	*C*1	6	38 7	123	*C*1	7	8 53
5 $b =$ 100	1 000	220	*C*1	7	28 2	214	*C*1	6	143 0	214	*C*1	7	26 8
5 $b =$ 50	3 000	390	*C*1	7	33 0	372	*C*1	6	122	372	*C*1	7	32 2

TABLE 2

Numerical Comparison of Broyden, Schubert and CUM, restart Every 6 Iterations

Problem	n	Broyden				Schubert				CUM			
		STOR	RSTP	ITER	TIME	STOR	RSTP	ITER	TIME	STOR	RSTP	ITER	TIME
1	1 000	31	$C0$	7	1 17	26	$C1$	5	1 63	26	$C1$	6	0 82
1	3 000	93	$C0$	7	3 52	78	$C1$	5	4 87	78	$C1$	6	2 45
1	5 000	155	$C0$	7	5 90	130	$C1$	5	8 20	130	$C1$	6	4 2
1	10 000	310	$C0$	7	13 2	260	$C1$	5	17 6	260	$C1$	6	8 48
1	15 000	465	$C0$	7	22 3	390	$C1$	5	29 0	390	$C1$	6	15 1
1	20 000	620	$C0$	7	32 2	520	$C1$	5	40 3	520	$C1$	6	22 3
2	1 000	69	$C0$	8	3 58	70	$C0$	7	7 77	63	$C0$	8	3 48
2	3 000	207	$C0$	8	10 8	210	$C0$	7	23 9	189	$C0$	8	10 5
2	5 000	335	$C0$	8	19 7	350	$C0$	7	45 4	315	$C0$	8	18 6
2	10 000	690	$C0$	8	48 4	700	$C0$	7	102	630	$C0$	8	46 4
3	1 000	51	$C0$	19	4 71	26	$C0$	12	4 89	31	$C0$	13	2 95
3	3 000	123	$C0$	13	9 19	78	$C0$	12	13 8	93	$C0$	13	8 64
3	5 000	205	$C0$	13	15 5	130	$C0$	12	24 2	155	$C0$	13	14 3
4	225	27	$C1$	4	1 05	27	$C0$	4	2 71	26	$C0$	5	
4	961	223	$C1$	4	13 1	222	$C1$	5	51 5	220	$C1$	5	13 9
5 $b =$ 15	1 000	60	$C0$	7	3 2	56	$C1$	6	6 55	55	$C0$	7	3 2
5 $b =$ 30	1 000	89	$C0$	7	6 84	85	$C1$	6	16 4	84	$C0$	7	6 82
5 $b =$ 50	1 000	127	$C0$	7	14 9	123	$C1$	6	39 7	122	$C0$	7	14 7
5 $b =$ 100	1 000	218	$C0$	7	49 3	214	$C1$	6	141 0	213	$C0$	7	50 1
5 $b =$ 50	3 000	384	$C0$	7	50 7	372	$C1$	6	122 0	369	$C0$	7	50 2

These experiments complement the ones reported by Martínez [17] for small-dimensional problems. They are much better than it could be predicted by the available theory. The existence of a local convergence result for CUM without restarts ($q = \infty$) may be conjectured. This result, as well as the corresponding superlinear convergence without restarts, is not easy to obtain, since general convergence theories [9, 18] are not applicable. We think that this conjecture deserves future research.

ACKNOWLEDGEMENTS

We acknowledge FINEP, CNPq, FAPESP and IBM for financial support given to this work.

We also acknowledge an anonymous referee for his thoughtful reading of the first version of this paper.

REFERENCES

[1] R. H. Bartels and G. H. Golub, *The Simplex Method of Linear Programming using LU decomposition,* Comm. ACM 12 (1969) 266-268.

[2] C. G. Broyden, *A class of methods for solving nonlinear simultaneous equations,* Math. Comp. 19 (1965) 577-593.

[3] C. G Broyden, *The convergence of an algorithm for solving sparse nonlinear systems,* Math. Comp. 25 (1971) 285-294.

[4] C. G. Broyden, J. E. Dennis and J. J. Moré, *On the local and superlinear convergence of quasi-Newton methods,* J. Inst. Math. Appl. 12 (1973) 223-246.

[5] J. E. Dennis, *Toward a unified convergence theory for Newton-like methods,* in L. B. Rall, ed., Nonlinear functional analysis and applications, Academic Press, New York, London, 1971, pp. 425-472.

[6] J. E. Dennis and J. J. Moré, *A characterization of superlinear convergence and its application to quasi-Newton methods,* Math. Comp. 28 (1974) 543-560.

[7] J. E. Dennis and R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations,* Prentice Hall, Englewood Cliffs, New Jersey, 1983

[8] J. E. Dennis and R. B. Schnabel, *A View of Unconstrained Optimization,* to appear in Handbook in Operations Research and Management Science, Vol. 1, Optimization, G L. Nemhauser, AHG Rinnooy Kan, M J. Tood, eds., North Holland, Amsterdam 1989.

[9] J. E. Dennis and H. F. Walker, *Convergence theorems for least-change secant update methods,* SIAM J. Numer Anal 18 (1981), 949-987.

[10] I. S. Duff, A. M. Erisman and J. K. Reid, *Direct methods for sparse matrices,* Clarendon Press, Oxford, 1986.

[11] A. George and E. Ng, *Symbolic factorization for sparse Gaussian elimination with partial pivoting,* SIAM J. Sci. Statist. Comput. 8 (1987), 877-898.

[12] G. H. Golub and Ch. F. Van Loan, *Matrix Computations,* John Hopkins, Baltimore, 1983.

[13] W. A. Gruver and E. Sachs, *Algorithmic methods in optimal control,* Pitman, Boston, London, Melbourne, 1981.

[14] L. V. Kantorovich and G. P. Akilov, *Functional analysis in normed spaces,* MacMillan, New York, 1964.

[15] T. Kato, *Perturbation theory for linear operators,* Springer Verlag, New York, 1966.

[16] A. Kolmogoroff and S. Fomin, *Elements of the Theory of Functions and Functional Analysis,* Izdat. Moscow Univ., Moscow, 1954.

[17] J. M. Martínez, *A quasi-Newton method with modification of one column per iteration,* Computing 33 (1984), 353-362.

[18] J. M. Martínez, *A new family of quasi-Newton methods with direct secant updates of matrix factorizations,* SIAM J. Numer. Anal. 27 (1990), 1034-1049.

[19] E S. MARWIL, *Convergence results for Schubert's method for solving sparse nonlinear equations,* SIAM J. Numer. Anal. 16 (1979), 588-604

[20] H. MATTHIES and G. STRANG, *The solution of nonlinear finite element equations,* Internat. J. Numer. Methods in Engrg. 14 (1979), 1613-1626.

[21] J. M. ORTEGA and W. C. RHEINBOLDT, *Iterative solution of nonlinear equations in several variables,* Academic Press, New York, 1970.

[22] E. SACHS, *Convergence rates of quasi-Newton algorithms for some nonsmooth optimization problems,* SIAM J. Control Optim. 23 (1985), 401-418

[23] E SACHS, *Broyden's method in Hilbert space,* Math. Programming 35 (1986), 71-82.

[24] L. K. SCHUBERT, *Modification of a quasi-Newton method for nonlinear equations with a sparse Jacobian,* Math. Comp 24 (1970), 27-30

[25] L. K. SCHUBERT, *An interval arithmetic approach for the construction of an almost globally convergence method for the solution of the nonlinear Poisson equation on the unit square,* SIAM J Sci. Statist. Comput 5 (1984), 427-452.

[26] H SCHWETLICK, *Numerische Losung nichtlinearer Gleichungen,* Berlin Deutscher Verlag der Wissenschaften, 1978

[27] Ph. L TOINT, *Numerical solution of large sets of algebraic nonlinear equations,* Math. Comp. 16 (1986), 175-189.