ELSEVIER

Journal of Computational and Applied Mathematics 76 (1996) 77–88

JOURNAL OF
COMPUTATIONAL AND
APPLIED MATHEMATICS

Partially Updated Switching-Method for systems of nonlinear equations [1]

S. Bellavia, M.G. Gasparo, M. Macconi *

Dipartimento di Energetica, Università di Firenze, via Cesare Lombroso 6/17, I-50134 Firenze, Italy

Abstract

A hybrid method for solving systems of n nonlinear equations is given. The method does not use derivative information and is especially attractive when good starting points are not available and the given system is expensive to evaluate. It is shown that, after a few steps, each iteration requires $(2k + 1)$ function evaluations where k, $1 \leqslant k \leqslant n$, is chosen so as to have an efficient algorithm. Global convergence results are given and superlinear convergence is established. Some numerical results show the numerical performance of the proposed method.

Keywords: Nonlinear systems; Damped secant method; Direct search method; Superlinear convergence; Efficiency

AMS classification: 65H10; 65H20

1. Introduction

We consider the numerical solution of systems of nonlinear equations

$$F(x) = 0, \tag{1}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is a given mapping. It is known that most common algorithms for solving (1) show severe difficulties to converge when starting away from a solution x^*. Then they are not very useful when a good initial guess is not available. This explains the great interest in global methods, i.e. iterative processes converging from a wide range of initial points [2]. Further, many times derivatives are very expensive and for many applications analytic derivatives are not available. Therefore there is an interest in algorithms where Jacobian information is not required.

Very recently, a globally convergent method avoiding explicit computation of the Jacobian was presented by the authors in [1]. The method given in [1] is a hybrid method where a slowly

[*] Corresponding author. Fax: ++39-55-4224137; e-mail: macconi@ingfi1.ing.unifi.it.

[1] Work supported by C.N.R., Progetto Finalizzato "Sistemi Informatici e Calcolo Parallelo", Sottoprogetto 1 and by M.U.R.S.T. 40%, Rome, Italy.

PII S0377-0427(96)00091-X

convergent global method is matched with a fast local one. The method was called Switching-Method to emphasize that at each iteration suitable tests automatically select which of the two schemes is to be used.

In this paper a modification of the Switching-Method is proposed. This modification is obtained by combining a direct search method with a damped secant one in the class of rank-k methods, where k is a fixed integer such that $1 \leqslant k \leqslant n$. The choice $k = n$ gives the original Switching-Method.

Since each iteration of the proposed modification involves the updating of k columns among n of the iteration matrix, this new iterative method will be called PUS-Method (Partially Updated Switching-Method).

It is shown that the PUS-Method is a globally convergent method yielding a finite difference approximation of the Jacobian. Further, after a finite number of iterations, the method requires $(2k + 1)$ function evaluations per iteration. Therefore, compared with the original method given in [1], partial updating of the iteration matrix yields a method with less computational cost of each iteration. Theoretical analysis of the asympotic properties of the proposed methods shows that the convergence is superlinear with k-depending rate. Since the convergence of the original method is quadratic, overall effectiveness of both methods is compared. The result is that the proposed PUS-Method is preferable for nonlinear systems for which the computational cost of the F-evaluation is at least of the order $O(n^2)$.

Numerical results confirm the effectiveness of the PUS-Method. All nonlinear systems in the collections [4, 5] were used for an actual comparison between the proposed modification and the original method. Some selected results are given to emphasize the reliability and the high efficiency of the PUS-Method in the numerical solution of computationally expensive problems.

2. The PUS-Method

Let k, $1 \leqslant k \leqslant n$, be a fixed integer. Each iteration of the PUS-Method can be viewed in terms of two different schemes: a direct search method where k coordinate directions are used at a time and a damped secant method where k columns of the iteration matrix are updated at a time. Suitable tests decide which of these two schemes is to be used at each iteration. The first scheme will be denoted as the kCD-Method where CD means "coordinate directions". The second one will be denoted as the kUC-Method where UC means ''updated columns".

In order to give a formal description of the PUS-Method we shall begin by describing the kCD-Method and the kUC-Method separately. The Euclidean vector norm and the spectral matrix norm will be used, both of which will be denoted by $\|\cdot\|$. Further, e_j will denote the jth unit coordinate vector and $m = \lceil n/k \rceil$ will be the smallest integer such that $n/k \leqslant m$, i.e. $m - 1 < n/k \leqslant m$.

The kCD-Method is a direct search method for the unconstrained minimization problem

$$\min_{x \in \mathbb{R}^n} f(x), \tag{2}$$

where $f : \mathbb{R}^n \to \mathbb{R}$. Starting from a given $x^{(0)} \in \mathbb{R}^n$, we construct a sequence $\{x^{(i)}\}$ such that $f(x^{(i+1)}) < f(x^{(i)})$ for $i \geqslant 0$ and such that each iterate $x^{(i+1)}$ is computed by comparing f-values at

$2k$ trial points around $x^{(i)}$. More precisely we define a trial set $T = \{j_1, \dots, j_k\}$ of distinct integers such that $1 \leqslant j_s \leqslant n$, for $s = 1, \dots, k$ and consider $2k$ trial points

$$\xi_j = x^{(i)} + \varepsilon^{(i)} e_j \quad \text{and} \quad \xi_{n+j} = x^{(i)} - \varepsilon^{(i)} e_j, \quad j \in T, \tag{3}$$

where the steplength $\varepsilon^{(i)} > 0$ is given. If some trial point gives a lower value than $x^{(i)}$ of the objective function, the iteration is successful. Otherwise the trial set T is changed and other different $2k$ trial points are tested. In the worst case, $2n$ trial points are visited without obtaining decrease in $f(x)$. In this case, the steplength $\varepsilon^{(i)}$ is reduced (halved) and new trial points are considered. The rule for defining the trial sets T is immaterial so long as all $2n$ trial points around $x^{(i)}$ are tried before halving $\varepsilon^{(i)}$.

The kUC-Method was initiated by Mukai [6] to solve nonlinear systems (1). Marking the current iterate $x^{(i)}$, the next approximation $x^{(i+1)}$ is given by

$$x^{(i+1)} = x^{(i)} + \lambda^{(i)} s^{(i)}, \tag{4}$$

where $s^{(i)}$ solves

$$H^{(i)} s^{(i)} = -F(x^{(i)}) \tag{5}$$

and the damping factor $\lambda^{(i)}$, $\lambda_{\min} \leqslant \lambda^{(i)} \leqslant 1$, is such that

$$\|F(x^{(i+1)})\|^2 \leqslant \theta \|F(x^{(i)})\|^2 \tag{6}$$

with fixed $\theta \in (0,1)$ and $\lambda_{\min} > 0$.

In (5) the iteration matrix $H^{(i)}$ differs from previous $H^{(i-1)}$ only in k columns. More precisely, if K_i is the set of k indices of columns to be updated and $\varepsilon^{(i)}$ is a given positive real number, the matrix $H^{(i)}$ is given by

$$H^{(i)} e_j = H^{(i-1)} e_j, \quad j \notin K_i, \tag{7}$$

$$H^{(i)} e_j = \pm [F(x^{(i)} \pm \varepsilon^{(i)} e_j) - F(x^{(i)})] / \varepsilon^{(i)}, \quad j \in K_i, \tag{8}$$

where, only for convenience, the sign plus is taken if

$$\|F(x^{(i)} + \varepsilon^{(i)} e_j)\| < \|F(x^{(i)} - \varepsilon^{(i)} e_j)\|, \tag{9}$$

otherwise the sign minus is chosen.

We remark that from both theoretical and practical points of view, the unit damping factor $\lambda^{(i)} = 1$ must be chosen whenever possible. So, at each iteration $\lambda^{(i)} = 1$ is tried. If for this value (6) is not satisfied, a bisection procedure is used to obtain an acceptable damping factor.

To explain the basic idea of the proposed hybrid method we remark that an iteration of kUC-Method is successful if the iteration matrix $H^{(i)}$ is invertible and (4) produces a point $x^{(i+1)}$ satisfying (6). If the iteration of kUC-Method fails to find $x^{(i+1)}$, then an iteration of the kCD-Method is applied to the minimization problem (2) with

$$f(x) = \tfrac{1}{2} \|F(x)\|^2. \tag{10}$$

It is possible to match the kUC-Method and the kCD-Method in such a way that all information gained by one scheme is exploited when a switch to the alternative method occurs. In this way we obtain the PUS-Method. More precisely, given $x^{(i)}$ we start with a fixed trial set T_1 by updating k columns of the iteration matrix according to (8) with $K_i = T_1$ and try a kUC-iteration. If this is not successful, the kCD-Method is used with initial trial set T_1 to check up on decrease in $\|F(x)\|$. If (10) does not decrease, new F-values must be computed. Such F-values are used to update the iteration matrix and try again kUC-Method. Formally each iteration step of the PUS-Method is described in the following way:

```
0. Given $x^{(i)}$, $\varepsilon^{(i)}$, $H^{(i-1)}$, $k$
1. Set $H^{(i)} = H^{(i-1)}$
2. For $l = 1,\dots,m$ do
     Define $T_l = \{j_1,\dots,j_k\}$
     For $j = j_1,\dots,j_k$ do
           Compute $\xi_j = x^{(i)} + \varepsilon^{(i)} e_j$ and $\xi_{n+j} = x^{(i)} - \varepsilon^{(i)} e_j$
           If $\|F(\xi_j)\| < \|F(\xi_{n+j})\|$, then
                 Set $\xi = \xi_j$ and $\rho = \varepsilon^{(i)}$
           else
                 Set $\xi = \xi_{n+j}$ and $\rho = -\varepsilon^{(i)}$
           endif
           Compute $H^{(i)} e_j = (F(\xi) - F(x^{(i)})/\rho$
     Try a $k$UC-iteration
     If $k$UC-iteration is successful, then
           Compute $\varepsilon^{(i+1)} = \min\{\varepsilon^{(i)}, \|x^{(i+1)} - x^{(i)}\|, \|F(x^{(i+1)})\|\}$
           Go to step 4
     else
           Compute $\xi$ s.t. $\|F(\xi)\| = \min\{\|F(\xi_j)\|, \|F(\xi_{n+j})\|\ , j \in T_l\}$
           If $\|F(\xi)\| < \|F(x^{(i)})\|$, then
                 Set $x^{(i+1)} = \xi$ and $\varepsilon^{(i+1)} = \varepsilon^{(i)}$
                 Go to step 4
           endif
     endif
3. Set $\varepsilon^{(i)} = \varepsilon^{(i)}/2$ and go to step 2
4. End of iteration
```

We remark that the updating of $\varepsilon^{(i)}$ used in the above algorithm should be suitably adjusted for problems that are badly scaled [2]. Further, in order to implement the PUS-Method one has to specify the rule to define the trial sets T_l and the value of the constant k. As to the trial sets, they can be arbitrarily chosen so long as each column of the iteration matrix is updated at least once in m iterations and all coordinate directions are considered before halving $\varepsilon^{(i)}$. The choice of the parameter k will be discussed in Section 4.

We point out that if $k = 1$ is used, the kCD-Method is related to the local variations method exploited by E. Polak in [9]. The Polak's method differs from the PUS-Method with $k = 1$ in the rule to visit the trial points and to update the steplength. Therefore, the PUS-Method with $k = 1$ and Polak's method generate different sequences of points $x^{(i)}$.

3. Convergence results

The convergence properties of the PUS-Method will be now stated under the following assumptions:

(i) Given a vector $x^{(0)} \in \Re^n$ the set $C(x^{(0)}) = \{x \in \mathbb{R}^n : \ \|F(x)\| \leqslant \|F(x^{(0)})\|\}$ is bounded and contained in a convex open set D.

(ii) $F(x)$ is twice continuously differentiable in D.

(iii) The Jacobian $J(x)$ of $F(x)$ is invertible.

(iv) $X = \{x^* \in C(x^{(0)}) \colon \ F(x^*) = 0\}$ contains a finite number of points.

From these assumptions it follows that for any $x^* \in X$ there exist positive constants L, γ, ρ such that for all x and y in the closed ball $B(x^*, \rho)$ we have

$$\|J(x) - J(y)\| \leqslant L\|x - y\| \tag{11}$$

and

$$\|F(x)\| \leqslant \gamma\|x - x^*\|. \tag{12}$$

Further, since $\nabla f(x) = J^{\mathrm{T}}(x)F(x)$, assumption (iii) ensures that X is the set of the critical points for (10).

The convergence of the kCD-Method and the global convergence of the PUS-Method can be obtained by slight modification of the proofs of Theorems 3.1 and 3.2 given in [1]. We state without proof the following results.

Theorem 3.1. *Under assumptions* (i)–(iv) *the sequence* $\{x^{(i)}\}$ *constructed by the kCD-Method converges to a point* $x^* \in X$.

Theorem 3.2. *Under assumptions* (i)–(iv) *the sequence* $\{x^{(i)}\}$ *constructed by the PUS-Method converges to a point* $x^* \in X$.

The next Theorem 3.3 states that it is possible that the first iterations are performed by the kCD-Method, but the switch over the kUC-Method occurs after a finite number of iterations. Moreover, for i sufficiently large, $x^{(i+1)}$ is obtained by the kUC-Method with unit damping factor $\lambda^{(i)}$ and the iteration matrix is a difference approximation of the Jacobian.

Theorem 3.3. *Let* $\{x^{(i)}\}$ *be a convergent sequence generated by the PUS-Method. Then, under assumptions* (i)–(iv), *there exists an integer* $M_1 > 0$ *such that for* $i \geqslant M_1$

$$x^{(i+1)} = x^{(i)} - H_i^{-1}F(x^{(i)}), \tag{13}$$

where $H_i^{-1} = [H^{(i)}]^{-1}$.

Proof. Since $\varepsilon^{(i)} \to 0$ and $x^{(i)} \to x^* \in X$ as $i \to \infty$, there exists $M > m - 1$ such that for $i > M$ we have $\varepsilon^{(i-m+1)} < \rho/2$ and $x^{(i-s)} \in B(x^*, \rho/2)$ for $s = 0, 1, \ldots, m-1$. Let $i > M$. By construction, for any

$j = 1,2,\ldots,n$ there exists an integer s, $0 \leqslant s \leqslant m-1$, such that

$$H^{(i)}e_j = \pm\frac{1}{\varepsilon^{(i-s)}}[F(x^{(i-s)} \pm \varepsilon^{(i-s)}e_j) - F(x^{(i-s)})]$$
$$= \int_0^1 J(x^{(i-s)} \pm t\varepsilon^{(i-s)}e_j)e_j\,\mathrm{d}t.$$

It follows that

$$\|H^{(i)}e_j - J(x^{(i)})e_j\| \leqslant \int_0^1 \|J(x^{(i-s)} \pm t\varepsilon^{(i-s)}e_j)e_j - J(x^{(i)})e_j\|\,\mathrm{d}t$$

and then, by using (11),

$$\|H^{(i)}e_j - J(x^{(i)})e_j\| \leqslant L[\|x^{(i)} - x^*\| + \|x^{(i-s)} - x^*\| + \tfrac{1}{2}\varepsilon^{(i-s)}]. \tag{14}$$

By the properties of the norms it follows that $\|H^{(i)} - J(x^{(i)})\| \to 0$ as $i \to \infty$. At this point, the result follows by reasoning totally analogous to that found in the proof of Theorem 3.3 of [1]. □

The order of convergence of the PUS-method will now be investigated. To this end, we recall that for any integer $q \geqslant 0$ the algebraic equation

$$t^{q+1} - t^q - 1 = 0$$

has a unique positive solution τ_q with the following properties: $1 < \tau_q \leqslant 2$, $\tau_0 = 2$, $\tau_q > \tau_{q+1}$ and $\tau_q \to 1$ as $q \to \infty$ [7].

Theorem 3.4. *The R-order of the PUS-method is at least* τ_{m-1}.

Proof. Assume that the sequence $\{x^{(i)}\}$ generated by the PUS-Method converges to a solution x^* of $F(x)=0$. Further, let $M > 0$ be an integer such that for $i \geqslant M$, $x^{(i-s)}$ is obtained by the kUC-Method for $s = 0,1,\ldots,m-1$. Then, by construction, $\varepsilon^{(i-s)} \leqslant \|F(x^{(i-s)})\|$ and from (12) and (14) we obtain for all $j = 1,\ldots,n$,

$$\|H^{(i)}e_j - J(x^{(i)})e_j\| \leqslant L[\|x^{(i)} - x^*\| + (1 + \tfrac{1}{2}\gamma)\ \|x^{(i-s)} - x^*\|].$$

Hence, the properties of the norms imply the existence of $\beta > 0$ such that

$$\|H^{(i)} - J(x^{(i)})\| \leqslant \beta \sum_{j=0}^{m-1} \|x^{(i-j)} - x^*\|.$$

Then, from the mean-value theorem and (11) it follows that for i sufficiently large

$$\|x^{(i+1)} - x^*\| = \|x^{(i)} - H_i^{-1}F(x^{(i)}) - x^*\|$$
$$\leqslant \|x^{(i)} - x^*\|\|H_i^{-1}\|[\|H^{(i)} - J(x^{(i)})\| + L\|x^{(i)} - x^*\|]$$
$$\leqslant \|H_i^{-1}\|\|x^{(i)} - x^*\|[\beta \sum_{j=0}^{m-1} \|x^{(i-j)} - x^*\| + L\|x^{(i)} - x^*\|].$$

Therefore,

$$\|x^{(i+1)} - x^*\| \leqslant \|x^{(i)} - x^*\| \sum_{j=0}^{m-1} \gamma_j \|x^{(i-j)} - x^*\|,$$

where $\gamma_0, \ldots, \gamma_{m-1}$ are suitable positive constants. Then, by the result 9.2.9 of [7], the R-order is at least τ_{m-1}. □

As one of the referees pointed out, our results hold with continuity of first derivates and condition (11) instead of the assumption of the existence of second derivatives. The convergence analysis under these assumptions is completely analogous to one given in [3] with some additional minor technical difficulties.

In conclusion, we saw that the PUS-method preserves global convergence property of the original Switching-Method. Further, the convergence is superlinear with k-depending rate and the quadratic convergence is obtained when $k = n$.

4. Implementation

In order to implement the PUS-Method we must decide how to choose the starting iteration matrix $H^{(0)}$ and the number k of columns updated at each iteration.

Concerning the choice of $H^{(0)}$ we remark that the global convergence of the PUS-Method is proved without any particular assumptions on $H^{(0)}$. In particular, $H^{(0)}$ is not required to be the Jacobian at $x^{(0)}$ or its approximation. Moreover, the starting matrix may be a singular matrix. In this case a number of function evaluations is required to obtain an invertible iteration matrix. In particular, if the null matrix $H^{(0)} = 0$ is used, $2n$ function evaluations are spent in this initial phase. On the other hand, if a nonsingular $H^{(0)}$ is used it is entirely possible that a number of function evaluations may be required to tempt the kUC-Method before any real progress is achieved. However, the PUS-Method recovers from a bad (singular) initial iteration matrix and automatically constructs a right approximation to the Jacobian.

Concerning the choice of the number k of columns updated at each iteration, we remark that the computational cost of each iteration depends on k. Following Ostrowsky [8], we consider as the measure of the efficiency of the method the quantity:

$$E(k) = L(k)/W(k),$$

where $L(k) = \log \tau_{m-1}$ and $W(k)$ is the limit of the amount of work required to perform one iteration. In the PUS-Method the computational cost of each iteration increases with k. On the other hand, the convergence rate decreases when k decreases, as it follows from Theorem 3.4. For these reasons, we are interested in finding a value $k < n$ such that the PUS-Method takes less computing time to verify a fixed stopping criterion. From Theorem 3.3 it follows that after a finite number of iterations the damping factor $\lambda^{(i)} = 1$ is used in (4). In this case, each iteration involves $(2k+1)$ F- evaluations and the solution of one linear algebraic system. Thus, the asymptotic amount of work $W(k)$ of the PUS-Method is given by $W(k) = (2k+1)C_F + C_A$ where C_F is the cost of one F-evaluation and C_A

the cost of the solution of (5). It seems reasonable to choose $k = k^*$ maximizing

$$E(k) = L(k)/[(2k+1)C_F + C_A]. \tag{15}$$

Concerning an estimation of C_F we note that in general, it is not feasible to count all the floating point operations involved by each F-evaluation. However, most of the important ones may be counted assuming that elementary arithmetic operations and elementary functions have each unitary cost. About C_A, assume that matrix factorization techniques are used for linear algebraic systems. Then, well-known estimates of the arithmetic cost C_A are available.

In our implementation of the PUS-Method the QR-factorization is used. In this case the theoretical approach based on maximization of (15) recommends the value $k^* = n$ for nonlinear systems with computational cost $C_F \leqslant n^2$. In order to find a minimal C_F for which the optimal k is less than n, we reasonably assume $n \geqslant 10$. In this case, the theoretical suggestion $k^* \leqslant \lceil n/2 \rceil$ is obtained for problems with $C_F \geqslant 2n^2$.

It is worth noting that approach (15) does not take into account the initial speed of the method. Numerical experiments involving problems with different computational cost C_F are essential to decide if this approach leads to a more computationally efficient algorithm than the choice $k = n$.

In Table 1 we give the k^*-values resulting from some pairs (C_F, n) used in the numerical experiments.

Table 1
Theoretical k^*-values when the QR-factorization is used

	$C_F = n^2$	$C_F = 2n^2$	$C_F = 3n^2$	$C_F = 4n^2$	$C_F = 5n^2$
n	k^*	k^*	k^*	k^*	k^*
10	10	5	5	5	2
20	20	10	5	5	5
40	40	20	10	10	10
60	60	30	20	15	15
80	80	40	27	20	16
100	100	50	25	25	20
200	200	100	67	50	40
400	400	200	134	100	80

	$C_F = 10n^2$	$C_F = 15n^2$	$C_F = 20n^2$	$C_F = 25n^2$	$C_F = 50n^2$
n	k^*	k^*	k^*	k^*	k^*
10	2	2	2	2	1
20	4	2	2	2	2
40	5	4	4	4	2
60	10	6	5	5	3
80	10	8	8	5	4
100	10	10	10	5	5
200	25	20	20	10	8
400	50	40	25	25	16

5. Some numerical results

In this section, we present some numerical results obtained in the solution of standard test problems taken from [4, 5]. All problems were solved with increasing dimension and several starting points. In particular, for each fixed n, we used $x^{(0)}=10^j x_s$, $j=0,1,2,3$ where x_s is the standard initial point. In this way we tested the robustness of the PUS-Method , i.e. its ability to solve problems from a wide range of initial points.

All computations were performed on a IBM Risc 6000 with the precision of about 16 decimal places. The initial steplength was $\varepsilon^{(0)} = 0.1\|x^{(0)}\|$ and $\theta = 0.975$ was used in (6). Further, $\lambda_{\min} = 0.125$ was used. In all numerical experiments $H^{(0)}$ was chosen as the null $n \times n$ matrix. In this way the algorithm was tested without any favor. Convergence was declared when either $\|F(x^{(i)})\| \leqslant 10^{-9}$ or $\|x^{(i+1)} - x^{(i)}\| \leqslant 10^{-9}\|x^{(i)}\| + 10^{-9}$. Failure was declared when convergence was not achieved after $I_{\max} = \max(20n/k, 500)$ iterations, when $F_{\max} = 500n$ F-evaluations were performed, when the steplength $\varepsilon^{(i)}$ was reduced below $\varepsilon_{\min} = 10^{-7}$. The QR-factorization was decided for numerical solution of linear algebraic systems and the theoretical approach given

Table 2
Some results for Problem 1

n	$x^{(0)}$	k	$\|F\|$	N_{CD}	N_{UC}	N_F	T_k/T_n
150	x_s	150	0.0	0	9	3938	1.0
		75	0.0	1	9	3217	1.2
		30	0.0	0	14	3225	2.4
		15	0.0	1	15	3147	4.1
		10	0.0	2	15	3195	5.7
		6	0.0	2	16	3475	9.1
	10^2x_s	150	0.5(−9)	0	2	603	1.0
		75	0.4(−9)	1	2	453	1.1
		30	0.4(−9)	4	2	363	1.6
		15	0.4(−9)	9	2	333	2.6
		10	0.3(−9)	14	2	323	3.6
		6	0.3(−9)	24	2	315	5.8
400	x_s	400	0.6(−13)	0	2	1603	1.0
		200	0.6(−13)	1	2	1203	1.2
		100	0.6(−13)	3	2	1003	1.8
		50	0.6(−13)	7	2	903	2.9
		25	0.6(−13)	15	2	853	5.1
		10	0.6(−13)	39	2	823	11.7
	10^2x_s	400	0.0	0	3	2404	1.0
		200	0.1(−9)	1	3	1604	1.1
		100	0.8(−10)	3	3	1204	1.4
		50	0.6(−10)	7	3	1004	2.1
		25	0.5(−10)	15	3	904	3.6
		10	0.5(−10)	39	3	844	7.9

in the previous section was used to decide the number k of columns updated at each iteration.

To have some measure of the overall efficiency of the PUS-Method , we considered the number N_{CD} of iterations performed by the kCD-Method, the number N_{UC} of iterations performed by the kUC-Method, the total number N_F of required F-evaluations and the ratio T_k/T_n, where T_k is the time of computation spent with the choice $k < n$ and T_n is the time of computation spent with $k=n$.

The accuracy of the PUS-Method was measured by the norm $\|F\|$ of the function value at the computed solution.

To show typical behavior of the PUS-Method we give some selected results obtained with the Extended Rosenbrock function, the Gheri–Mancino function and a set of problems with variable computational cost obtained by repeated evaluations of the extended Rosenbrock function.

Problem 1. The extended Rosenbrock function in n variables is given by

$$\begin{aligned} F_{2i-1}(x) &= 10(x_{2i} - x_{2i-1}^2), \\ F_{2i}(x) &= 1 - x_{2i-1}, \end{aligned} \qquad i = 1,\ldots,\tfrac{1}{2}n.$$

This very popular test problem was solved for $n = 50$, 100, 150,..., 400.

The PUS-Method converged to $x=(1,1,\ldots,1)^{\mathrm{T}}$ for all used starting vectors $x^{(0)}=10^j x_s$, $j=0,1,2,3$ where $x_s = (-1.2, 1.0, \ldots, -1.2, 1.0)^{\mathrm{T}}$.

The estimated computational cost of each F-evaluation is $C_F = 5n$ and the choice $k = n$ is recommended for the PUS-Method. This is confirmed in the practice. It turned out that when k decreases more iterations are performed, less F-evaluations are generally computed but the minimum time of computation spent to reach the convergence occurs for $k = n$. These facts are shown in the Table 2 for some values of n and $x^{(0)}$.

Problem 2. The Gheri–Mancino function in n variables is given by

$$\begin{aligned} F_i(x) = {} & 14nx_i + (i - n/2)^3 \\ & + \sum_{j=1}^{i-1} \alpha_{ij}(\sin^5(\ln \alpha_{ij}) + \cos^5(\ln \alpha_{ij})) \\ & + \sum_{j=i+1}^{n} \alpha_{ij}(\sin^5(\ln \alpha_{ij}) + \cos^5(\ln \alpha_{ij})), \quad i = 1,\ldots,n, \end{aligned}$$

where $\alpha_{ij} = \sqrt{x_j^2 + i/j}$.

We solved this problems with increasing values of n. The used dimensions range from $n = 10$ to $n = 200$.

The problem $F(x) = 0$ was solved in [4] where the computed solution is given for $n = 50$. For $n=50$, we obtained a final approximation which coincides with the result in [4] up to five significant digits at least.

The used standard initial guess was $x_s = -[(c_1+c_2)/2c_1c_2]F(0)$ where $c_1 = 20n-6$ and $c_2 = 8n+6$.

The estimated computational cost of one F-evaluation is $C_F = 14n^2 - 7n$. It was verified that the optimal values of k are those of Table 1 with $C_F = 15n^2$.

Table 3
Some results for the Problem 2

$x^{(0)}$	n	k^*	k	$\|F\|$	N_{CD}	N_{UC}	N_F	T_k/T_n
x_s	10	2	10	0.3(−10)	0	3	64	
			2	0.4(−11)	1	4	37	0.2
	20	2	20	0.8(−13)	0	4	165	
			2	0.5(−10)	1	5	62	0.4
	30	3	30	0.5(−12)	0	4	245	
			3	0.6(−9)	1	5	90	0.4
	40	4	40	0.6(−11)	0	4	325	
			4	0.5(−9)	1	5	118	0.4
	50	5	50	0.3(−9)	0	4	405	
			5	0.4(−9)	1	6	157	0.4
$10x_s$	10	2	10	0.1(−13)	0	5	106	
			2	0.7(−12)	2	6	47	0.6
	20	2	20	0.8(−13)	0	5	206	
			2	0.2(−11)	5	7	72	0.4
	30	3	30	0.7(−12)	0	5	306	
			3	0.8(−9)	5	7	104	0.3
	40	4	40	0.9(−12)	0	5	406	
			4	0.1(−8)	5	6	127	0.3
	50	5	50	0.2(−11)	0	6	607	
			5	0.4(−8)	5	6	157	0.3
10^2x_s	10	2	10	0.7(−14)	0	5	106	
			2	0.4(−9)	2	6	47	0.5
	20	2	20	0.8(−13)	0	6	247	
			2	0.5(−8)	5	6	67	0.3
	30	3	30	0.5(−12)	0	6	367	
			3	0.4(−11)	5	7	104	0.3
	40	4	40	0.9(−12)	0	6	487	
			4	0.1(−9)	5	7	136	0.3
	50	5	50	0.2(−11)	0	6	607	
			5	0.8(−9)	5	7	168	0.3

The obtained numerical results show the good performance of the proposed PUS-Method . The number of F-evaluations is essentially reduced when $k = k^*$ is used instead of $k = n$ and the time of computation T_k is always more than halved with respect to T_n.

In Table 3 we give the results obtained with $n = 10, 20, \ldots, 50$. Analogous behavior was observed for all used values of n.

Problem 3. A set of problems with variable computational cost was defined by repeated evaluations of the Extended Rosenbrock function.

We used the same n and $x^{(0)}$ of Problem 1 and we generated problems with $C_F = pn^2$ with $p = 5, 10, 15, \ldots, 50$.

The values N_{CD}, N_{UC}, N_F and $\|F\|$ are independent on the variation of C_F. On the contrary, the computational time is very sensitive to C_F and it was verified that when $k = k^*$ is used a lot of

Table 4
Effectiveness of solving Problem 3 with $k = k^*$

C_F	n	k^*	T_{k^*}/T_n	C_F	n	k^*	T_{k^*}/T_n	C_F	n	k^*	T_{k^*}/T_n
$5n^2$	100	20	0.7	$10n^2$	100	10	0.7	$15n^2$	100	10	0.6
	150	30	0.7		150	19	0.7		150	15	0.6
	200	40	0.6		200	25	0.6		200	20	0.6
	250	50	0.5		250	36	0.5		250	25	0.4
	300	60	0.6		300	43	0.6		300	30	0.6
	400	80	0.5		400	50	0.4		400	40	0.4
$20n^2$	100	10	0.6	$25n^2$	100	5	0.6	$50n^2$	100	5	0.6
	150	10	0.6		150	10	0.6		150	6	0.6
	200	20	0.6		200	10	0.6		200	8	0.6
	250	18	0.4		250	18	0.4		250	10	0.4
	300	25	0.6		300	20	0.6		300	12	0.5
	400	25	0.4		400	25	0.4		400	16	0.4

time is saved compared with the time spent with $k = n$. As an example we show in Table 4 the results obtained with $x^{(0)} = 10^2 x_s$. Analogous behavior was observed for all the used $x^{(0)} = 10^j x_s$, for $j = 1, 2, 3$. In these cases, the time of computation was about halved when $k = k^*$ was chosen.

With the initial point $x^{(0)} = x_s$ such important saving of computational time was not observed but different choices of $\varepsilon^{(0)}$ have been successfully used to this end.

Acknowledgements

The authors thank the referees for detailed, careful and helpful suggestions.

References

[1] S. Bellavia, M.G. Gasparo and M. Macconi, A Switching-Method for nonlinear systems, *J. Comput. Appl. Math.*, to appear.

[2] J.E. Dennis Jr. and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (Prentice-Hall, Englewood Cliffs, NJ, 1983).

[3] M.G. Gasparo, Sulla convergenza di un procedimento basato sul metodo delle secanti modificato, *Calcolo* **XXI** (1984) 75–89.

[4] G. Gheri and O.G. Mancino, A significant example to test methods for solving systems of non linear equations, *Calcolo* **8** (1971) 107–113.

[5] J.J. Moré, B.S. Garbow and K.E. Hillstrom, Testing unconstrained optimization software, *ACM Trans. Math. Software* **7** (1981) 17–41.

[6] H. Mukai, Parallel algorithms for solving systems of nonlinear equations, *Comput. Math. Appl.* **7** (1981) 235–250.

[7] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables* (Academic Press, New York, 1970).

[8] A.M. Ostrowsky, *Solution of Equations and Systems of Equations* (Academic Press, New York, 1960).

[9] E. Polak, A globally converging secant method with applications to boundary value problems, *SIAM J. Numer. Anal.* **11** (1974) 529–537.