# SOLVING NONLINEAR SYSTEMS OF EQUATIONS BY MEANS OF QUASI-NEWTON METHODS WITH A NONMONOTONE STRATEGY [1]

[2]

**Keywords:** Nonlinear systems, quasi-Newton methods, Trust regions, nonmonotone strategies.

**Abstract**

A nonmonotone strategy for solving nonlinear systems of equations is introduced. The idea consists of combining efficient local methods with an algorithm that reduces monotonically the squared norm of the system in a proper way. The local methods used are Newton's method and two quasi-Newton algorithms. Global iterations are based on recently introduced box-constrained minimization algorithms. We present numerical experiments.

## 1 INTRODUCTION

Given $F : I\!R^n \to I\!R^n, F = (f_1, \ldots, f_n)^T$, our aim is to find solutions of

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

We assume that $F$ is well defined and has continuous partial derivatives on an open set of $I\!R^n$. $J(x)$ denotes the Jacobian matrix of partial derivatives of $F(x)$. We are mostly interested in problems where $n$ is large and $J(x)$ is *structurally sparse*. This means that most entries of $J(x)$ are zero *for all $x$ in the domain of $F$*.

The package *Nightingale* has been developed at the Department of Applied Mathematics of the University of Campinas for solving large-scale nonlinear systems of equations. See [24], [25]. In *Nightingale*, Newton's method and seven quasi-Newton methods are implemented. Our numerical experience in the last few years indicate that the best quasi-Newton methods included in the package are Broyden's good method [1] and the method based on the Column-Updating formula ([38], [45], [24]). Of course, Newton's

[2]Institute of Mathematics, University of Campinas, IMECC-UNICAMP, CP 6065, 13081-970 Campinas SP, Brazil

method must also be kept within a package for solving nonlinear systems, since it tends to be the most efficient method when the cost of solving linear systems is not very high.

In this paper, we adopt the point of view that derivatives are available, either by hand calculation or by automatic differentiation. See [26] and references therein. So, Newton's method with analytic derivatives is fully implementable. We use quasi-Newton methods with rank-one corrections, so that the linear algebra associated to a quasi-Newton iteration is, generally, very small compared with the one associated to the Newton iteration.

Local convergence properties of Newton and quasi-Newton methods are well known. Briefly speaking, under suitable hypotheses, Newton's method has local quadratic convergence while most quasi-Newton methods have local superlinear convergence. Fortunately, the convergence region of these methods is usually much larger than the one predicted by local convergence theorems, so they can succeed even if the initial guess is not very close to the solution. However, in many cases their robustness can be improved if modifications that turn them "globally convergent" are introduced. Usually, global convergence is obtained by means of the introduction of a merit function for which a monotone decreasing behavior is required. Nevertheless, it has been observed by some authors ([31],[32], [50]),[55]) that, even in the unconstrained minimization problem, it is worthwhile allowing the merit function to be increased at some iterations in order to improve the overall convergence behavior. In this research we implement a tolerant non-monotone strategy that fits well with local quasi-Newton steps.

As it is usual, we use the merit function given by

$$f(x) = \frac{1}{2}\|F(x)\|^2, \tag{2}$$

where $\|\cdot\| = \|\cdot\|_2$ from now on. When a solution of (1) exists, to solve this problem is equivalent to find a global minimizer of $f$.

However, the decision of merely using a method to minimize $f$, in order to solve (1), is not satisfactory by the following reasons:

(a) Minimization algorithms applied to (2) make a poor use of the structure of the nonlinear system. For example, they generally need second order information for obtaining quadratic convergence, while Newton's method for nonlinear systems is quadratically convergent using only first derivatives.

(b) Sometimes, efficient local methods converge rapidly to a solution but $\{f(x^k)\}$ does not decrease monotonically. This phenomenon cannot occur at the final iterations of a locally superlinear convergent method but it has been observed in other situations. A well known extreme case is when a rank-one secant method is applied to a nonsingular linear system. According to Gay's theorem [21], finite convergence occurs in at most $2n$ iterations. However, the behavior of $f(x)$ at these iterations is highly nonmonotone. If the method is modified (for example, using line searches) in order to force a monotone behavior of $f$ the performance becomes extremely poor. A popular example of rapid convergence accompanied by nonmonotone behavior is given by Rosenbrock's system $10(x_2 - x_1^2) = 0$, $1 - x_1 = 0$. For most initial points, Newton's local method converges in two iterations but at the first iteration the norm of $F$ is increased. On the other hand, if a monotone method is used, many iterations are wasted following the classical narrow valley that characterizes the level sets of this function. When the iterates are in a proper neighborhood of the solution, the local Newton step satisfies decrease conditions and it is accepted by any reasonable monotone strategy. However, this neighborhood is, in practice, very small, and local acceptance of the local Newton step is not observed unless the iterate is very close to the solution. Of course, Rosenbrock's function is merely a simple model for badly scaled problems.

(c) Many efficient minimization algorithms converge to stationary points of $f$ that are not necessarily global minimizers. Moreover, local minimizers of $f$ are, sometimes, powerful attractors of the iterates. As it is well known, a local non-global minimizer of $f$ is, in general, not only a poor estimate of a solution of (1) but also a very bad initial point for initiating any process for solving the nonlinear system.

Some authors ([3], [2], [15], [37]) (see also [46] for nonsmooth systems) developed globalization procedures based on the function (2) that are not proper minimization procedures but are based on monotone decreasing of the function $f$ (or some other merit function with similar properties). These algorithms make a better use of information than minimization algorithms but they do not prevent convergence to undesirable points when singular Jacobians are present.

By these reasons, it is desirable to give a chance to the local method before calling the monotone algorithm. Different alternatives have been suggested with this purpose (see [31], [32]). Here, we describe a strategy that combines local and monotone steps. A related strategy (with different acceptance criteria) has been used in [14] for some overdetermined systems coming from inverse problems.

This paper is organized as follows. In Section 2 we describe a strategy for combining *local* and *global* iterations that aims to take advantage of the good features of both. In Section 3 we describe the *local* iterations that are used in our package. In Section 4 we describe the *global* iterations. We give numerical experiments in Section 5. Final remarks are given in Section 6.

## 2 THE TOLERANT STRATEGY

Let us define "ordinary iterations" and "special iterations". By an ordinary iteration we understand an iteration produced by any local method. A special iteration is an iteration produced by a monotone algorithm applied to $f$. We define, for all $k = 0, 1, 2, \ldots$,

$$a^k = \text{Argmin}\,\{f(x^0), \ldots, f(x^k)\}. \tag{3}$$

For completeness, we define $f(a^k) = BIG \geq f(x^0)$ if $k < 0$. Ordinary and special iterations are combined by the following algorithm.

**Algorithm 2.1.** Initialize $k \leftarrow 0$, FLAG $\leftarrow 1$. Let $q \geq 0$ be an integer, $\gamma \in (0, 1)$.

*Step 1.* If FLAG $= 1$, obtain $x^{k+1}$ using an ordinary iteration.
Else, obtain $x^{k+1}$ using a special iteration.

*Step 2.* If

$$f(x^{k+1}) \leq \gamma f(a^{k-q}) \tag{4}$$

then set FLAG $\leftarrow 1$.
Else, re-define $x^{k+1} \leftarrow a^{k+1}$ and set FLAG $\leftarrow -1$.
Set $k \leftarrow k + 1$ and go to Step 1. $\odot$

If the test (4) is satisfied infinitely many times, then there exists a subsequence of $\{x^k\}$ such that $\lim_{k \to \infty} \|F(x^k)\| = 0$. So, if the sequence $\{x^k\}$ is bounded, we will be able to find a solution of (1) up to any prescribed accuracy. Conversely, if (4) does not hold for all $k \geq k_0$, then all the iterations from the $k_0$-th on will be special, and the convergence properties of the sequence will be those of the monotone algorithm. The parameters $\gamma$ and $q$ measure the degree of tolerance in the local algorithm. If $q$ is large we admit a large number of iterations without enough progress. On the other hand, $\gamma$ says what we mean by "enough progress". The closer the real number $\gamma \in (0,1)$ is to the unity, the more tolerant the algorithm becomes.

## 3 ORDINARY ITERATIONS

We use three different ordinary iterations in our study: Newton, Broyden and Column-Updating. Newton's method is the most widely used algorithm for solving nonlinear systems of equations.

An iteration of Newton's method is described by

$$J(x^k)s^k = -F(x^k), \tag{5}$$

$$x^{k+1} = x^k + s^k. \tag{6}$$

At each iteration, we must compute the Jacobian $J(x^k)$ and solve the linear system (5). Using modern techniques of automatic differentiation we can compute $F(x)$ and $J(x)$ in a reliable and economical way. In *Nightingale* , we use the sparse linear solver of George and Ng [22]. The George-Ng method performs the $LU$ factorization with partial pivoting of a sparse matrix $A$ using a *static data structure* defined before beginning numerical computations. In Newton's method we solve a sequence of linear systems with the same structure, so, the *symbolic phase* that defines the data structure is executed only once.

The system (5) has a unique solution if and only if $J(x^k)$ is nonsingular. If the Jacobian is singular, the iteration must be modified. Moreover, if $J(x^k)$ is nearly singular, it is also convenient to modify the iteration in order to prevent numerical instability. In the *Nightingale* package, when a very small pivot, relatively to the size of the matrix, occurs, it is replaced by a nonzero scalar whose modulus is sufficiently large. Moreover, nearly singular or ill-conditioned matrices usually cause very large increments $s^k$. So, $\|s^k\|$ must also be controlled. As many other computer algorithms, *Nightingale* normalizes the stepsize by

$$s^k \leftarrow \min\{1, \frac{\beta}{\|s^k\|}\}s^k.$$

where $\beta$ is a parameter given by the user. Of course, these modifications do not preserve the property of minimizing the local model $\|J(x^k)s + F(x^k)\|$. They are, however, a simple and generally useful way (see [47]) of avoiding very large steps while preserving the already computed (or partially computed) $LU$ factorization of the Jacobian.

If $F$ has continuous partial derivatives, $J(x)$ satisfies a Lipschitz condition and is nonsingular at a solution $x^*$, then the sequence of iterates generated by Newton's method is well defined if $x^0$ is close enough to $x^*$ and converges quadratically. See [49], [9].

We call quasi-Newton methods those methods for solving (1) whose general form is

$$x^{k+1} = x^k - B_k^{-1} F(x^k).$$

Newton's method belongs to this family. Most quasi-Newton methods use less expensive iterations than Newton, but their convergence properties are not very different. In general, quasi-Newton methods avoid either the necessity of computing derivatives, the necessity of solving a full linear system per iteration or both tasks. Broyden's method and the Column-Updating method are *secant* methods, that is, quasi-Newton methods that satisfy the secant equation:

$$B_{k+1}(x^{k+1} - x^k) = F(x^{k+1}) - F(x^k).$$

In both methods, writing $s^k = x^{k+1} - x^k$ and $y^k = F(x^{k+1}) - F(x^k)$, we have that

$$B_{k+1} = B_k + \frac{(y^k - B_k s^k)(z^k)^T}{(z^k)^T s^k} \tag{7}$$

where

$$z^k = x^{k+1} - x^k$$

for Broyden's method and

$$z^k = e^{j_k}, \ \ |(e^{j_k})^T s^k| = \|s^k\|_\infty$$

for the Column-Updating method. (The set $\{e^1, \ldots, e^n\}$ is the canonical basis of $\mathbb{R}^n$.)

Applying the Sherman-Morrison formula to (7) ([23], p. 51]) we obtain

$$B_{k+1}^{-1} = B_k^{-1} + \frac{(s^k - B_k^{-1} y^k)(z^k)^T}{(z^k)^T B_k^{-1} y^k} B_k^{-1}. \tag{8}$$

Formula (8) shows that $B_{k+1}^{-1}$ can be obtained from $B_k^{-1}$ using $O(n^2)$ floating point operations in the dense case. Moreover,

$$B_{k+1}^{-1} = (I + u^k (z^k)^T) B_k^{-1},$$

where $u^k = (s^k - B_k^{-1} y^k)/(z^k)^T B_k^{-1} y^k$, so

$$B_k^{-1} = (I + u^{k-1}(z^{k-1})^T) \dots (I + u^0 (z^0)^T) B_0^{-1}, \tag{9}$$

for $k = 1, 2, 3 \dots$

Formula (9) is used when $n$ is large. In this way, the computer time of iteration $k$ is $O(kn)$ plus the computer time of computing $B_0^{-1} F(x^k)$. A special device introduced in [13] permits the implementation of Broyden's method storing only one additional $n-$vector per iteration. See, also, [12] and [35]. If $k$ is large the process must be periodically restarted taking $B_k \approx J(x^k)$.

Broyden's method is a particular case of the family of Least Change Secant Update (LCSU) methods ([8], [9], [10], [40], [42]), which include many algorithms that are useful for particular structures. See [7], [36], [27], [28], [29], [30], [54], [34], [39], etc.

With the same hypotheses that imply local convergence of Newton's method, for a good enough initial Jacobian approximation $B_0$, LCSU algorithms are well defined, converge to a solution $x^*$ and the rate of convergence is q-superlinear. See [4], [40], [41], [42].

After some years of practical experience, we verified that the Column-Updating method has essentially the same behavior as Broyden's method, in terms of reliability and number of iterations. Many times, the Column-Updating method uses less computer time than Broyden because the iteration is slightly less expensive. However, the Column-Updating method is not an LCSU method, and local and superlinear convergence cannot be proved using the classical techniques. Local convergence results for the Column-Updating method have been given in [38] and [45].

# 4 SPECIAL ITERATIONS

Special iterations can be defined by using any convergent monotone method described in the literature. For large-scale problems, we use a strategy based on trust regions derived from the inexact Newton approach. See [6].

Essentially, this strategy is the one given in [18] for bound constrained minimization and it is also related to the box constrained algorithms of [5]. The strategy is described by the following algorithm.

**Algorithm 4.1.** Assume that $\Delta_{\min} > 0, \alpha \in (0,1)$ are given independently of the iteration $k$. Define $\psi_k(x) = \|F(x^k) + J(x^k)(x - x^k)\|^2$ , $\Delta_k \geq \Delta_{\min}$ .

*Step 1.* Compute an approximate minimizer $\overline{x}$ of $\psi_k(x)$ on the box $\|x - x^k\|_\infty \leq \Delta_k$ such that $\psi_k(\overline{x}) \leq \psi_k(x_Q^k)$, $x_Q^k$ is the projection of $x^k - 2J(x^k)^T F(x^k)/M_k$ on the box and $M_k \geq 2\|J(x_k)\|_1 \|J(x_k)\|_\infty$.

*Step 2.* If

$$\|F(\overline{x})\|^2 \leq \|F(x^k)\|^2 + \alpha(\psi_k(\overline{x}) - \psi_k(x^k)) \tag{10}$$

define $x^{k+1} = \overline{x}$. Else, choose $\Delta_{\text{new}} \in [0.1\|\overline{x} - x^k\|, 0.9\Delta_k]$, replace $\Delta_k$ by $\Delta_{\text{new}}$ and go to Step 1. $\odot$

Most of the work of Algorithm 4.1 is concentrated on the approximate solution of the quadratic problem

$$\begin{array}{ll} \text{Minimize} & \psi_k(x) \\ \text{subject to} & \|x - x^k\|_\infty \leq \Delta_k. \end{array} \tag{11}$$

For this problem, algorithms based on combinations of Krylov subspace methods with gradient projection strategies are currently preferred. We use a variation of the algorithm introduced in [16]. In such work the Krylov iterations are classical conjugate-gradient iterations [33]. In the present modification, we use GMRES iterations [51] in the interior of the box (11), which turned out to be less sensitive than C-G iterations to ill-conditioning of $J(x^k)$. The approximate solution of (11) is defined as a point that satisfies $\psi_k(\overline{x}) \leq \psi_k(x_Q^k)$ where, in addition, the norm of the projected gradient is less than $0.1\|J(x^k)^T F(x^k)\|$. We also choose: $\Delta_{\min} = 100\|x^0\|$, initial choice of $\Delta \equiv \Delta_0 = 1000\|x^0\|$, $\Delta_{\text{new}} = 0.5\|\overline{x} - x^k\|$, and $\Delta_{k+1} = \max\{\Delta_{min}, 4\Delta_k\}$.

The convergence properties of Algorithm 4.1 are a direct consequence of the results of [18]. Every limit point $x^*$ of a sequence $\{x^k\}$ generated by this algorithm satisfies $J(x^*)^T F(x^*) = 0$. Therefore, $x^*$ is a solution of (1) if $J(x^*)$ is nonsingular.

Other recent methods based on the inexact Newton approach with global convergence properties were given by Deuflhard [11], Eisenstat and Walker

[15], Brown and Saad [3] and Lukšan [37]. An advantage of special iterations based on box trust regions is that they can be easily adapted to situations where we have natural bounds for the solution of (1). In fact, it is common in practical problems that the pursued solutions belong to a region of $I\!R^n$, usually described by bounds on the variables. This is the case of many physical constraints or positivity constraints induced by optimization problems (see, for example, [19]).

# 5 NUMERICAL EXPERIMENTS

Algorithm 2.1 was implemented combining two procedures: ordinary iterations are quasi-Newton and special iterations are performed using Algorithm 4.1. Both codes are written in FORTRAN 77 and ran in a SUN Sparc Station 2. We use single precision at ordinary iterations on the experiments of Subsection 5.1, and double precision elsewhere. The results of the global methods are sensitive to the choice of the parameters $BIG$, $\gamma$ and $q$, but, on average, many triplets $(BIG, \gamma, q)$ are equally efficient. We used $BIG = f(x_0)$, $\gamma = 0.5$ and $q = 3$ in all the experiments.

Subsections 5.1, 5.2 and 5.3 present problems for which Newton's method without any globalization strategy converges in many cases. If one strongly believes in the fact that pure local Newton method converges, in most cases, only when the initial approximation is close enough to the solution, then one could claim that these three set of problems are not really difficult. However, as far as 1979, Moré and Cosnard [47] showed that, in small to medium systems, quadratically convergent methods (without globalization) are more efficient than globalized quasi-Newton methods based on monotone decrease of a norm of $F$. They also observed that a good procedure for controlling the steplength in critical cases plays the role of stabilizing the process, without monotone decrease of $\|F(x)\|$. The numerical results obtained for the first three set of problems seem to confirm Moré and Cosnard's conclusions. Moreover, these results suggest that global iterations should be used only when they are clearly necessary, if the underlying local method is robust enough. In some cases, as in the convection-diffusion problem for $C = 100$, forcing decrease of $\|F(x)\|$ is useful for final convergence. In other cases, as the cavity problem, homotopic procedures seem to be much more efficient for obtaining global convergence. In subsection 5.4 a set of difficult problems is presented where some globalization strategy turns out to be essential for achieving convergence. We compare our tolerant global strategy with a monotone trust-region algorithm.

### 5.1 Lukšan's set of problems

Our first set of problems is described by Lukšan in [37] and was used by this author to test monotone inexact Newton methods. In addition to the tests with $n = 100$ (reported in [37]), we also ran the problems with $n = 1000$. The description of the problems is given in [37].

Among the several methods implemented in *Nightingale* , we chose Newton's method, Broyden's method and the Column-Updating method to perform the ordinary iterations.

The stopping criteria used are the following: in the local methods, convergence is declared at $x^k$ whenever $\|F(x^k)\|_2 < \sqrt{n}10^{-5}$ and divergence is declared if either the number of iterations performed is greater than 100 or $\|F(x^k)\|_\infty > 10^{20}$. In the globalized methods, convergence at iteration $k$ is declared if either $\|F(x^k)\|_2 < \sqrt{n}10^{-5}$ or $\Delta_k < 10^{-4}$. In the former we have convergence to a solution to (1) whereas in the latter $x^k$ is a "strangled point", stationary for problem (2) but not necessarily a solution of (1). Divergence is declared as in the local methods. The step control parameter is $\beta = 1000 \max\{1, \|x^0\|_\infty\}$.

Our test results are summarized in Tables 1, 2 and 3 where, respectively, we present the performance of Newton's method, Broyden's method and the Column-Updating method both in their local and global versions. In Broyden's method and in the Column-Updating method, the local version consists of performing the first iteration of Newton's method, followed by iterations of the corresponding method. In the global version, every cycle of local iterations is executed with a newtonian initialization. In our tables we adopted the following notation to present the numerical results: (IT, NF, T) for the local methods and (IT (N,G), EVALF, NF, T) for the global ones, where IT represents the total number of iterations performed; NF = $\|F(\overline{x})\|_2/\sqrt{n}$, where $\overline{x}$ is the vector obtained in the last iteration; T is the total time spent (in seconds) in the numerical phase; N is the total number of Newton iterations; G is the total number of special iterations and EVALF is the total number of evaluations of $f(x)$. When $\|F(\overline{x})\|_\infty > 10^{20}$, NF is $\|F(\overline{x})\|_\infty$.

We observe that the symbolic phase is performed twice for each problem since we have two different dimensions ($n = 100$ and $n = 1000$). For the smaller dimension, it lasts 0.01 seconds on average. For $n = 1000$, the average time spent in the symbolic phase is 0.1 seconds.

As it can be observed from the numerical results, in this set of problems our implementation of Newton's local method with step control converges in 31 out of the 34 tests performed. Broyden's and the Column-Updating method are far less robust, although most cases of nonconvergence of these methods are solved by their globalized version. Broadly speaking, when a local method generates a convergent sequence, the corresponding global version has a certain disadvantage if running time is considered. That occurs because special iterations are more expensive, and even though the number of iterations performed can be reduced from the local version to the global one, the total running time is certainly larger.

## 5.2 Driven cavity flows

A different set of experiments was performed using a classical problem in fluid mechanics, the *cavity problem*. See, for example, [52]. The problem is to find $\psi : \Omega \to I\!\!R$, $\Omega = [0,1] \times [0,1]$ such that

$$\Delta^2 \psi + Re[\frac{\partial \psi}{\partial x_1} \frac{\partial}{\partial x_2} \Delta \psi - \frac{\partial \psi}{\partial x_2} \frac{\partial}{\partial x_1} \Delta \psi] = 0 \text{ in } \Omega$$

with the boundary conditions

$$\psi = 0 \text{ on } \partial\Omega$$

$$\frac{\partial \psi}{\partial n}(x_1, 0) = \frac{\partial \psi}{\partial n}(0, x_2) = \frac{\partial \psi}{\partial n}(1, x_2) = 0,$$

$$\frac{\partial \psi}{\partial n}(x_1, 1) = 1.$$

We used the discretization scheme of [52] with a $63 \times 63$ grid. The nonlinearity of the problem increases as $Re$, the Reynolds number, increases. We solved the problem for 45 equally spaced Reynolds numbers, from $Re = 0$ to $Re = 11000$. We used the solution obtained for each Reynolds number $Re$ as initial point for the problem defined by $Re + 250$. In our computer, the cost of a Newton iteration was, approximately 199 seconds. Broyden and the Column-Updating method wasted, roughly, 11 seconds per iteration. Spurious solutions, which are likely to appear with coarser grids (see [52]) did not occur all along the calculations. Special iterations were not necessary

Table 1: Lukšan's problems: Newton's Method.

| Problem | Dimension | Newton Local Version | Newton Global Version |
|---|---|---|---|
| 1 | n=100 | (100, 0.2E-2, 1.02) | (11(9,2), 22, 0.4E-5, 2.15) |
| | n=1000 | (100, 0.7E-3, 8.82) | (24(15,9), 53, 0.9E-5, 305.4) |
| 2 | n=100 | (13, 0.3E-6, 0.08) | (100(7,93), 1107, 0.7E-4, 18.35) |
| | n=1000 | (13, 0.9E-6, 0.71) | (100(16, 84), 268, 0.4E-3, 431.82) |
| 3 | n=100 | (2, 0.2E-5, 0.02) | (2(2,0), 3, 0.2E-5, 0.02) |
| | n=1000 | (2, 0.6E-6, 0.2) | (2(2,0), 3, 0.6E-6, 0.2) |
| 4 | n=100 | (8, 0.2E-5, 0.08) | (6(5,1), 12, 0.2E-6, 0.19) |
| | n=1000 | (8, 0.6E-6, 0.66) | (8(8,0), 9, 0.6E-6, 0.65) |
| 5 | n=100 | (7, 0.6E-5, 0.09) | (7(7,0), 8, 0.6E-5, 0.09) |
| | n=1000 | (100, 0.1E-4, 11.85) | (100(6,94), 744, 2.0, 2509.6) |
| 6 | n=100 | (9, 0.8E-5, 0.05) | (9(9,0), 10, 0.8E-5, 0.07) |
| | n=1000 | (9, 0.6E-5, 0.55) | (9(9,0), 10, 0.6E-5, 0.53) |
| 7 | n=100 | (11, 0.6E-5, 0.08) | (11(11,0), 12, 0.6E-5, 0.09) |
| | n=1000 | (11, 0.2E-5, 0.69) | (11(11,0), 12, 0.2E-5, 0.72) |
| 8 | n=100 | (18, 0.8E-6, 0.2) | (19(12,7), 38, 0.2E-5, 2.93) |
| | n=1000 | (18, 0.3E-5, 1.62) | (12(11,1), 24, 0.1E-5, 12.2) |
| 9 | n=100 | (14, 0.9E-6, 0.20) | (13(11,2), 23, 0.5E-6, 1.09) |
| | n=1000 | (14, 0.3E-6, 1.83) | (13(11,2), 24, 0.2E-6, 11.13) |
| 10 | n=100 | (4, 0.2E-6, 0.04) | (4(4,0), 5, 0.2E-6, 0.04) |
| | n=1000 | (4, 0.2E-6, 0.5) | (4(4,0), 5, 0.2E-6, 0.45) |
| 11 | n=100 | (2, 0.0, 0.01) | (2(2,0), 3, 0.0, 0.01) |
| | n=1000 | (2, 0.0, 0.07) | (2(2,0), 3, 0.0, 0.07) |
| 12 | n=100 | (10, 0.6E-5, 0.08) | (10(10,0), 11, 0.6E-5, 0.08) |
| | n=1000 | (10, 0.6E-5, 0.61) | (10(10,0), 11, 0.6E-5, 0.61) |
| 13 | n=100 | (11, 0.4E-5, 0.05) | (11(10,1), 16, 0.3E-5, 0.14) |
| | n=1000 | (11, 0.4E-5, 0.42) | (10(9,1), 17, 0.9E-5, 1.10) |
| 14 | n=100 | (3, 0.3E-5, 0.03) | (3(3,0), 4, 0.3E-5, 0.03) |
| | n=1000 | (3, 0.3E-5, 0.18) | (3(3,0), 4, 0.3E-5, 0.18) |
| 15 | n=100 | (5, 0.1E-6, 0.07) | (5(5,0), 6, 0.1E-6, 0.08) |
| | n=1000 | (5, 0.1E-6, 0.76) | (5(5,0), 6, 0.1E-6, 0.75) |
| 16 | n=100 | (7, 0.6E-6, 0.07) | (7(7,0), 8, 0.6E-6, 0.06) |
| | n=1000 | (8, 0.1E-5, 0.48) | (8(8,0), 9, 0.1E-5, 0.50) |
| 17 | n=100 | (3, 0.9E-5, 0.04) | (3(3,0), 4, 0.9E-5, 0.04) |
| | n=1000 | (3, 0.4E-5, 0.18) | (3(3,0), 4, 0.4E-5, 0.16) |

Table 2: Lukšan's problems: Broyden's Method.

| Problem | Dimension | Broyden Local Version | Broyden Global Version |
|---|---|---|---|
| 1 | n=100 | (100, 0.3E-1, 1.53) | (13(2,1), 15, 0.7E-5, 1.31) |
| | n=1000 | (100, 0.3E+9, 13.8) | (19(2,1), 21, 0.9E-5, 22.4) |
| 2 | n=100 | (9, 0.6, 0.04) | (100(3, 93), 1107, 0.7E-4, 18.1) |
| | n=1000 | (9, 1.79, 0.3) | (100(8, 84), 268, 0.4E-3, 437.4) |
| 3 | n=100 | (3, 0.1E-5, 0.03) | (3(1,0), 4, 0.1E-5, 0.03) |
| | n=1000 | (2, 0.5E-5, 0.14) | (2(1,0), 3, 0.5E-5, 0.15) |
| 4 | n=100 | (35, 0.7E+24, 0.35) | (6(2,1), 12, 0.7E-5, 0.23) |
| | n=1000 | (29, 0.1E+22, 2.1) | (9(2,1), 17, 0.1E-5, 2.27) |
| 5 | n=100 | (46, 0.7E+21, 0.56) | (35(3,2), 45, 0.5E-5, 1.4) |
| | n=1000 | (25, 0.1E+25, 1.89) | (100(4, 89), 716, 2.0, 2155.9) |
| 6 | n=100 | (78, 0.7E-5, 1.0) | (16(2,1), 18, 0.5E-5, 0.14) |
| | n=1000 | (100, 0.5E+10, 13.2) | (18(1,1), 20, 0.8E-5, 1.16) |
| 7 | n=100 | (100, 3.66, 1.58) | (100(8,82), 962, 0.7E-1, 42.16) |
| | n=1000 | (100, 1.09, 13.3) | (100(10,78), 964, 0.2E-1, 356.0) |
| 8 | n=100 | (100, 0.1E-1, 1.58) | (17(2,1), 19, 0.3E-5, 0.3) |
| | n=1000 | (100, 0.3E-2, 14.1) | (24(5,10), 62, 0.5E-6, 41.9) |
| 9 | n=100 | (100, 0.2, 1.74) | (21(5,6), 46, 0.8E-5, 4.21) |
| | n=1000 | (100, 0.1E+10, 15.0) | (24(4,4), 32, 0.4E-5, 11.6) |
| 10 | n=100 | (9, 0.3E-5, 0.06) | (9(1,0), 10, 0.3E-5, 0.08) |
| | n=1000 | (8, 0.6E-5, 0.44) | (8(1,0), 9, 0.6E-5, 0.44) |
| 11 | n=100 | (3, 0.3E-5, 0.01) | (17(6,8), 62, 0.0, 0.83) |
| | n=1000 | (4, 0.0, 0.12) | (16(7,8), 66, 0.0, 7.85) |
| 12 | n=100 | (14, 0.7E-5, 0.08) | (14(1,0), 15, 0.7E-5, 0.09) |
| | n=1000 | (14, 0.7E-5, 0.65) | (14(1,0), 15, 0.7E-5, 0.68) |
| 13 | n=100 | (3, 0.2E+39, 0.01) | (15(3,2), 21, 0.9E-5, 0.13) |
| | n=1000 | (3, 0.2E+39, 0.1) | (15(3,2), 23, 0.6E-5, 1.45) |
| 14 | n=100 | (6, 0.5E-5, 0.05) | (6(1,0), 7, 0.5E-5, 0.03) |
| | n=1000 | (6, 0.2E-5, 0.23) | (6(1,0), 7, 0.2E-5, 0.24) |
| 15 | n=100 | (12, 0.6E-5, 0.09) | (12(1,0), 13, 0.6E-5, 0.09) |
| | n=1000 | (10, 0.6E-5, 0.7) | (10(1,0), 11, 0.6E-5, 0.72) |
| 16 | n=100 | (19, 0.1E-5, 0.12) | (12(2,1), 14, 0.4E-6, 1.48) |
| | n=1000 | (100, 0.18E+6, 13.4) | (22(3,2), 25, 0.4E-5, 30.8) |
| 17 | n=100 | (6, 0.4E-5, 0.04) | (6(1,0), 7, 0.4E-5, 0.04) |
| | n=1000 | (6, 0.2E-5, 0.22) | (6(1,0), 7, 0.2E-5, 0.23) |

Table 3: Lukšan's problems: Column - Updating Method.

| Problem | Dimension | Column -Updating Local Version | Column -Updating Global Version |
|---|---|---|---|
| 1 | n=100 | (48, 0.9E-5, 0.37) | (16(2,1), 18, 0.8E-5, 1.32) |
| | n=1000 | (100, 0.1E+9, 8.35) | (19(2,2), 22, 0.3E-5, 39.4) |
| 2 | n=100 | (8, 0.5, 0.05) | (100(3,93), 1107, 0.7E-4, 18.1) |
| | n=1000 | (8, 1.57, 0.25) | (100(8, 84), 268, 0.4E-3, 434.1) |
| 3 | n=100 | (3, 0.2E-5, 0.03) | (3(1,0), 4, 0.2E-5, 0.03) |
| | n=1000 | (2, 0.7E-5, 0.15) | (2(1,0), 3, 0.7E-5, 0.14) |
| 4 | n=100 | (63, 0.1E+25, 0.57) | (7(2,1), 13, 0.3E-5, 0.25) |
| | n=1000 | (53, 0.1E+23, 3.46) | (10(2,1), 18, 0.9E-6, 2.27) |
| 5 | n=100 | (44, 0.2E+29, 0.44) | (31(3,2), 43, 0.2E-5, 1.37) |
| | n=1000 | (42, 0.1E+39, 2.91) | (100(5,78), 736, 1.4, 1836.8) |
| 6 | n=100 | (100, 0.1E-3, 1.01) | (14(2,1), 16, 0.5E-5, 0.12) |
| | n=1000 | (100, 0.1E-3, 7.81) | (13(1,1), 15, 0.9E-5, 0.78) |
| 7 | n=100 | (100, 15.8, 1.01) | (100(6,86), 951, 0.7E-1, 42.2) |
| | n=1000 | (100, 13.1, 7.79) | (29(5,8), 39, 0.3E-5, 14.6) |
| 8 | n=100 | (100, 0.14E+1, 1.02) | (16(3,3), 25, 0.1E-6, 2.37) |
| | n=1000 | (100, 0.1E+1, 8.59) | (16(4,4), 27, 0.3E-5, 26.2) |
| 9 | n=100 | (100, 0.5, 1.06) | (21(3,3), 30, 0.3E-5, 1.60) |
| | n=1000 | (100, 0.3, 9.41) | (24(3,3), 34, 0.7E-5, 17.7) |
| 10 | n=100 | (9, 0.4E-5, 0.04) | (9(1,0), 10, 0.4E-5, 0.04) |
| | n=1000 | (9, 0.4E-5, 0.41) | (9(1,0), 10, 0.4E-5, 0.45) |
| 11 | n=100 | (3, 0.0, 0.02) | (17(6,8), 62, 0.0, 0.83) |
| | n=1000 | (3, 0.0, 0.08) | (15(6,8), 64, 0.0, 7.72) |
| 12 | n=100 | (14, 0.7E-5, 0.08) | (14(1,0), 15, 0.7E-5, 0.07) |
| | n=1000 | (14, 0.7E-5, 0.53) | (14(1,0), 15, 0.7E-5, 0.51) |
| 13 | n=100 | (3, 0.2E+39, 0.01) | (15(3,2), 21, 0.5E-5, 0.18) |
| | n=1000 | (3, 0.2E+39, 0.09) | (15(3,2), 23, 0.8E-5, 1.43) |
| 14 | n=100 | (6, 0.7E-5, 0.02) | (6(1,0), 7, 0.7E-5, 0.04) |
| | n=1000 | (5, 0.7E-5, 0.18) | (5(1,0), 6, 0.7E-5, 0.18) |
| 15 | n=100 | (12, 0.5E-5, 0.09) | (8(2,1), 10, 0.3E-5, 0.11) |
| | n=1000 | (12, 0.4E-5, 0.77) | (8(2,1), 10, 0.6E-6, 1.18) |
| 16 | n=100 | (35, 0.1E-5, 0.26) | (17(2,1), 19, 0.2E-5, 2.05) |
| | n=1000 | (100, 0.3E+8, 7.94) | (28(3,6), 31, 0.2E-5, 110.4) |
| 17 | n=100 | (6, 0.4E-5, 0.04) | (6(1,0), 7, 0.4E-5, 0.04) |
| | n=1000 | (6, 0.1E-5, 0.20) | (6(1,0), 7, 0.1E-5, 0.19) |

in this set of problems, so local and global algorithms coincide. The results (number of iterations, computer time) are given in Table 4. The convergence criterion $\|F(x_k)\|_\infty \leq 10^{-10}$ was satisfied in all the experiments. We used the same step control parameter of subsection 5.1. As a summary, we observed that Newton solved the 45 problems in 473 minutes, while Broyden and the Column-Updating method used approximately 165 minutes to perform all the calculations.

Walker [56] reported the results of applying his inexact Newton method with backtracking to cavity problems using a different approach. A fast biharmonic solver was used as preconditioner. He intended the solution of the problems for $Re = 100$, 500 and 1000 starting with the initial approximation $u_0(x) = 0$ in the three cases. His method converged using 7 outer iterations for $Re = 100$ and using 12 outer iterations for $Re = 500$. For $Re = 1000$ the inexact Newton method failed. Backtracking was never used along these calculations.

We also ran our methods for the cavity problems with the null initial approximation and the convergence criterion $\|F(x)\|_\infty \leq 10^{-10}$. The results are given in Table 5. For each method we report $(n_1, n_2, n_3, time)$, the number of local iterations, the number of global iterations, the number of functional evaluations and the computer time. The symbol (*) indicates that the method did not converge after 100 iterations.

### 5.3 Convection-diffusion problem

We consider the partial differential equation given in [35], (p. 109):

$$-\Delta u + Cu(u_x + u_y) = g$$

in $[0,1] \times [0,1]$, with $u \equiv 0$ on the boundary. The function $g$ is such that the discretization of

$$10xy(1-x)(1-y)\ \exp\ (x^{4.5}) \tag{12}$$

is a solution of the discretized problem, using central differences. We used, as in [35], a $31 \times 31$ grid, the initial approximation $u_0 \equiv 0$ and the stopping criterion:

$$\|F(x^k)\|_2/31 \leq 0.1(1/32)^2[\|F(x^0)\|_2/31] + 0.1(1/32)^2.$$

We considered two cases: $C = 20$ and $C = 100$. We used the step control parameter $\beta$ as in 5.1. The results are given in Table 6. The meaning of $n_1, n_2, n_3$ and $time$ are the same as in Table 5.

The convection-diffusion problem with $C = 100$ starting with $x^0 \equiv (0)$ function is difficult. We can see that the local versions of Newton's method and the quasi-Newton methods do not converge after 100 iterations for all the methods. However, the tolerant strategy works very well in this case, forcing final convergence of all the algorithms.

Table 4: Cavity problems up to $Re = 11000$.

| Reynolds | Newton | Broyden | Column-Updating |
|---|---|---|---|
| 0 | ( 1 , 188.60 ) | ( 1 , 188.32 ) | ( 1 , 189.80 ) |
| 250 | ( 5 , 942.56 ) | ( 23 , 264.70 ) | ( 22 , 258.69 ) |
| 500 | ( 5 , 756.39 ) | ( 13 , 230.61 ) | ( 13 , 230.06 ) |
| 750 | ( 5 , 754.04 ) | ( 8 , 214.96 ) | ( 8 , 213.83 ) |
| 1000 | ( 5 , 754.02 ) | ( 6 , 207.83 ) | ( 7 , 210.76 ) |
| 1250 | ( 3 , 572.77 ) | ( 6 , 210.82 ) | ( 6 , 210.44 ) |
| 1500 | ( 3 , 585.68 ) | ( 5 , 211.31 ) | ( 5 , 210.68 ) |
| 1750 | ( 3 , 594.76 ) | ( 5 , 214.57 ) | ( 5 , 214.14 ) |
| 2000 | ( 3 , 598.95 ) | ( 4 , 212.97 ) | ( 4 , 212.49 ) |
| 2250 | ( 3 , 602.59 ) | ( 5 , 216.72 ) | ( 5 , 216.61 ) |
| 2500 | ( 3 , 607.52 ) | ( 5 , 219.01 ) | ( 5 , 219.47 ) |
| 2750 | ( 3 , 608.14 ) | ( 5 , 218.28 ) | ( 5 , 218.83 ) |
| 3000 | ( 3 , 613.12 ) | ( 5 , 219.74 ) | ( 5 , 219.83 ) |
| 3250 | ( 3 , 616.41 ) | ( 5 , 221.08 ) | ( 5 , 221.17 ) |
| 3500 | ( 3 , 615.55 ) | ( 4 , 218.13 ) | ( 4 , 218.35 ) |
| 3750 | ( 3 , 619.21 ) | ( 4 , 218.88 ) | ( 5 , 222.19 ) |
| 4000 | ( 3 , 619.82 ) | ( 4 , 219.20 ) | ( 4 , 218.74 ) |
| 4250 | ( 3 , 620.91 ) | ( 4 , 219.86 ) | ( 4 , 219.96 ) |
| 4500 | ( 3 , 621.30 ) | ( 4 , 219.82 ) | ( 4 , 220.03 ) |
| 4750 | ( 3 , 624.95 ) | ( 4 , 223.28 ) | ( 4 , 220.47 ) |
| 5000 | ( 3 , 621.63 ) | ( 4 , 223.61 ) | ( 4 , 220.64 ) |
| 5250 | ( 3 , 625.65 ) | ( 4 , 221.28 ) | ( 4 , 220.89 ) |
| 5500 | ( 3 , 627.07 ) | ( 4 , 221.57 ) | ( 4 , 221.15 ) |
| 5750 | ( 3 , 628.12 ) | ( 4 , 222.34 ) | ( 4 , 221.90 ) |
| 6000 | ( 3 , 627.15 ) | ( 4 , 221.64 ) | ( 4 , 220.98 ) |
| 6250 | ( 3 , 628.21 ) | ( 4 , 222.50 ) | ( 4 , 221.65 ) |
| 6500 | ( 3 , 631.79 ) | ( 4 , 222.61 ) | ( 4 , 222.69 ) |
| 6750 | ( 3 , 631.67 ) | ( 4 , 223.25 ) | ( 4 , 222.57 ) |
| 7000 | ( 3 , 629.86 ) | ( 3 , 219.63 ) | ( 3 , 219.23 ) |
| 7250 | ( 3 , 632.56 ) | ( 3 , 219.41 ) | ( 3 , 219.19 ) |
| 7500 | ( 3 , 630.92 ) | ( 3 , 220.57 ) | ( 3 , 220.19 ) |
| 7750 | ( 3 , 645.42 ) | ( 3 , 220.21 ) | ( 3 , 220.86 ) |
| 8000 | ( 3 , 642.68 ) | ( 3 , 219.93 ) | ( 3 , 220.46 ) |
| 8250 | ( 3 , 633.28 ) | ( 3 , 221.29 ) | ( 3 , 220.16 ) |
| 8500 | ( 3 , 638.68 ) | ( 3 , 220.83 ) | ( 3 , 220.29 ) |
| 8750 | ( 3 , 632.14 ) | ( 3 , 219.56 ) | ( 3 , 220.55 ) |
| 9000 | ( 3 , 642.20 ) | ( 3 , 219.56 ) | ( 3 , 220.92 ) |
| 9250 | ( 3 , 644.13 ) | ( 3 , 221.24 ) | ( 3 , 221.05 ) |
| 9500 | ( 3 , 644.02 ) | ( 3 , 221.61 ) | ( 3 , 220.83 ) |
| 9750 | ( 3 , 636.24 ) | ( 3 , 221.19 ) | ( 3 , 221.18 ) |
| 10000 | ( 3 , 637.02 ) | ( 3 , 220.98 ) | ( 3 , 220.73 ) |
| 10250 | ( 3 , 635.66 ) | ( 3 , 221.59 ) | ( 3 , 221.42 ) |
| 10500 | ( 3 , 636.71 ) | ( 4 , 224.97 ) | ( 4 , 224.55 ) |
| 10750 | ( 3 , 636.46 ) | ( 4 , 225.12 ) | ( 4 , 224.85 ) |
| 11000 | ( 3 , 635.67 ) | ( 4 , 224.71 ) | ( 4 , 224.43 ) |

Table 5: Additional experiments with the cavity problem.

| Method | $Re = 100$ | $Re = 500$ | $Re = 1000$ |
|---|---|---|---|
| Newton local | 4, 0, 5, 269 | 8, 0, 9, 521 | (*) |
| Newton global | 4, 0, 5, 269 | 8, 0, 9, 521 | (*) |
| Broyden local | 24, 0, 25, 99 | (*) | (*) |
| Broyden global | 24, 0, 25, 99 | 18, 3, 26, 325 | (*) |
| Column-Updating local | 26, 0, 27, 101 | (*) | (*) |
| Column-Updating global | 8, 2, 13, 180 | 14, 4, 25, 438 | (*) |

Table 6: Numerical results for the convection-diffusion problem.

| Method | $C = 20$ | $C = 100$ |
|---|---|---|
| Newton local | 2, 0, 3, 6.9 | (*) |
| Newton global | 2, 0, 3, 6.9 | 5, 14, 150, 239.1 |
| Broyden local | 8, 0, 9, 5.2 | (*) |
| Broyden global | 8, 0, 9, 5.2 | 10, 23, 269, 380. |
| Column-Updating local | 6, 0, 7, 4.6 | (*) |
| Column-Updating global | 6, 0, 7, 4.6 | 9, 23, 268, 380. |

### 5.4 A set of difficult problems

The set of problems considered in this section are *difficult* in the sense that the purely local Newton method, given by

$$x_{k+1} = x_k - J(x_k)^{-1}F(x_k),$$

does not converge, for different reasons, independently of the number of iterations allowed. The divergence is characterized by an unbounded increase of the norm of the iterates or oscillation of the generated sequence, but never because of singularity of matrix $J(x_k)$. For these problems we compare our methods with the tolerant strategy against a standard monotone globalization method: the Trust-Region method described in [15] as Algorithm TR. The parameters chosen for TR were $\overline{\delta}_0 = 100$, $t = 0.1$, $u = 0.1$, $\theta_{min} = 0.1$, $\theta_{max} = 0.5$. When the trust region radius $\delta_k$ needs to be increased we choose the new one in the interval $[\delta_k, 10\delta_k]$. The trust region subproblems are solved using a Moré-Sorensen-like algorithm (MS) for determining the Lagrange multiplier [48]. Observe that $[0.1\delta_k, 0.5\delta_k]$ and $[\delta_k, 10\delta_k]$ play the role of large interval tolerances for stopping MS. The problems are described below.

**Problem D1.** (Modified Rosenbrock)
$n$ even,

$$\begin{aligned} f_{2i-1}(x) &= \frac{1}{1+\exp(-x_{2i-1})} - 0.73 \\ f_{2i}(x) &= 10(x_{2i} - x_{2i-1}^2) \end{aligned},$$

$i = 1, \ldots, n/2$. Initial point: $(-1.8, -1, -1.8, -1, \ldots)^T$.

**Problem D2.** (Augmented Rosenbrock)
$n$ multiple of 4,

$$\begin{aligned} f_{4i-3}(x) &= 10(x_{4i-2} - x_{4i-3}^2) \\ f_{4i-2}(x) &= 1 - x_{4i-3} \\ f_{4i-1}(x) &= 1.25x_{4i-1} - 0.25x_{4i-1}^3 \\ f_{4i}(x) &= x_{4i} \end{aligned},$$

$i = 1, \ldots, n/4$. Initial point: $(-1.2, 1, -1, 20, -1.2, 1, -1, 20, \ldots)^T$.

**Problem D3.** (Powell badly scaled)
$n$ multiple of 2,

$$\begin{array}{rcl} f_{2i-1}(x) & = & 10^4 x_{2i-1} x_{2i} - 1 \\ f_{2i}(x) & = & \exp(-x_{2i-1}) + \exp(-x_{2i}) - 1.0001 \end{array},$$

$i = 1, \ldots, n/2$. Initial point : $(0, 100, 0, 100, \ldots)$.

**Problem D4.** (Augmented Powell badly scaled)
$n$ multiple of 3,

$$\begin{array}{rcl} f_{3i-2}(x) & = & 10^4 x_{3i-2} x_{3i-1} - 1 \\ f_{3i-1}(x) & = & \exp(-x_{3i-2}) + \exp(-x_{3i-1}) - 1.0001 \\ f_{3i}(x) & = & \varphi(x_{3i}) \end{array},$$

$i = 1, \ldots, n/3$, where

$$\varphi(t) = \begin{cases} \dfrac{t}{2} - 2, & \text{if} \quad t \leq -1 \\ \dfrac{1}{1998}(-1924 + 4551t + 888t^2 - 592t^3), & \text{if} \quad t \in [-1, 2] \\ \dfrac{t}{2} + 2, & \text{if} \quad t \geq 2 \end{cases}.$$

Initial point: $(0, 1, -4, 0, 1, -4, \ldots)^T$.

**Problem D5.** (Tridimensional valley)
$n$ multiple of 3,

$$\begin{array}{rcl} f_{3i-2}(x) & = & (c_2 x_{3i-2}^3 + c_1 x_{3i-2}) \exp(\frac{-x_{3i-2}^2}{100}) - 1 \\ f_{3i-1}(x) & = & 10(\sin(x_{3i-2}) - x_{3i-1}) \\ f_{3i}(x) & = & 10(\cos(x_{3i-2}) - x_{3i}) \end{array},$$

where

$$c_1 = 1.003344481605351, \quad c_2 = -3.344481605351171 \times 10^{-3},$$

$i = 1, \ldots, n/3$. Initial point: $(-4, 1, 2, 1, 2, 1, 2, \ldots)^T$.

**Problem D6.** (Shifted and augmented trigonometric function with an Euclidean sphere - modification of system given in [53]).

$n$ variable,

$$f_i(x) = n - 1 - \sum_{j=1}^{n-1} \cos(x_j - 1) + i(1 - \cos(x_i - 1)) - \sin(x_i - 1),$$

$i = 1, \ldots, n-1$ and

$$f_n(x) = \sum_{j=1}^{n} x_j^2 - 10000.$$

Initial point: $(0, \ldots, 0)$.

**Problem D7.** (Diagonal of three variables premultiplied by a quasi-orthogonal matrix)

$n$ multiple of 3,

$$\begin{array}{rcl} f_{3i-2}(x) & = & 0.6x_{3i-2} + 1.6x_{3i-1}^3 - 7.2x_{3i-1}^2 + 9.6x_{3i-1} - 4.8 \\ f_{3i-1}(x) & = & 0.48x_{3i-2} - 0.72x_{3i-1}^3 + 3.24x_{3i-1}^2 - 4.32x_{3i-1} - x_{3i} \\ & & +0.2x_{3i}^3 + 2.16 \\ f_{3i}(x) & = & 1.25x_{3i} - 0.25x_{3i}^3 \end{array},$$

$i = 1, \ldots, n/3$. Initial point: $(50, 0.5, -1, 50, 0.5, -1, \ldots)^T$.

**Problem D8.** (Diagonal of three variables premultiplied by an orthogonal matrix, combined with inverse trigonometric function)

$n$ multiple of 3

$$
\begin{aligned}
f_{3i-2}(x) &= 64(x_{3i-2} + x_{3i-1} + x_{3i}) - 0.64 + 0.48\text{atan}(x_{3i}) \\
&\quad +0.60(c_1 + c_2 x_{3i-1} + c_3 x_{3i-1}^2 + c_4 x_{3i-1}^3) \\
f_{3i-1}(x) &= 0.48 - 48(x_{3i-2} + x_{3i-1} + x_{3i}) + 0.36\text{atan}(x_{3i}) \ , \\
&\quad +0.80(c_1 + c_2 x_{3i-1} + c_3 x_{3i-1}^2 + c_4 x_{3i-1}^3) \\
f_{3i}(x) &= 0.60 - 60(x_{3i-2} + x_{3i-1} + x_{3i}) + 0.80\text{atan}(x_{3i})
\end{aligned}
$$

where

$$
\begin{array}{rclrcl}
c_1 &=& 13.901020408163270000 & c_2 &=& -1.4056122448979600000 \\
c_3 &=& -2.2183673469387760000 & c_4 &=& -0.27704081632653060000
\end{array}
$$

$i = 1, \ldots, n/3$. Initial point: $(10, -5.223, -1.393, 10, -5.223, -1.393, \ldots)^T$.

**Problem D9.** (Quadratics and nonlinear equation)

$Q_i \in I\!R^{(n-1)\times(n-1)}$, $i = 1, \ldots, n-1$ and $b_i \in I\!R^{n-1}$, $i = 1, \ldots, n-1$ have random elements between $-1$ and 1. We define

$$f_i(x) = \frac{1}{2} x^T Q_i x + b_i^T x$$

for $i = 1, \ldots, n-1$, and

$$f_n(x) = \text{atan}(x_1 + \ldots + x_n)$$

The initial point is $(1, 10, 100, 1000, 1, 10, 100, 1000, \ldots)^T$.

**Problem D10.** (Bratu)

This problem is defined by

$$-\Delta u - 50e^u = g \text{ in } \Omega \equiv [0,1] \times [0,1], \tag{13}$$

with $u = 0$ on the boundary and the source $g$ chosen in such a way that (12) is a solution of the discretized problem. We discretize by centered differences on a $15 \times 15$ grid. See [2]. As initial approximation we used $u_0 \equiv 0$.

The behavior of Algorithm TR and the global tolerant methods based on Newton, Broyden and Column-Updating are given in Table 7. Algorithm

TR was stopped when $\|F(x_k)\|_2 \leq 10^{-4}$ or when $\|\nabla f(x_k)\|_2^2 \leq 10^{-5}$. The global tolerant methods were stopped when $\|F(x_k)\|_2 \leq 10^{-7}$ (the reason for this tighter choice is the small number of iterations employed in most cases by these methods). In both algorithms the maximum allowed number of iterations was 100 and whenever the trust region radius ($\Delta_k$) was less than $10^{-8}$ the algorithms stopped at a local minimizer of the merit function $f$. For this set of problems the step control parameter is $\beta = 100$. We report total number of iterations, number of functional evaluations and the final value of $\|F(x)\|_2$ obtained, with the number of global iterations between parentheses in the case of the tolerant methods. Computer time of TR iterations is much larger than computer time of the other algorithms iterations, since an average of 2.5 Cholesky factorizations are used for each functional evaluation.

The numerical results presented on Table 7 show that 12 out of the 24 problems converged to the solution of system (1) for all the methods used, that is, for the monotone and the three tolerant strategies. For 3 problems none of the strategies managed to escape from a local minimizer of the merit function. For the remaining 9 problems, algorithm TR always stopped at a local solution, while the globalized Newton's method converged to the solution on (1) in 7 of these problems and each of the quasi-Newton globalizations found the global solution of 5 of these problems. Summarizing, algorithm TR converged to the global solution in 50% of the tests, Newton in 79%, Broyden in 71% and Column-Updating in 71%, which gives a slight advantage to Newton's globalized strategy.

In problem D6 solved by the tolerant Newton's method, the step control played the role of globalizing the generated sequence, so that it was not necessary to perform any special iteration. Comparing the two globalized quasi-Newton strategies, we have essentially the same pattern of success and failures, except for problem D7 ($n = 3$ and $n = 33$), where the behavior is complementary. It is interesting to notice that a quasi-Newton tolerant strategy was able to succeed when both TR and Newton failed. That is the case of problems D7 ($n = 33$) and D9. All the reported experiments were run with the parameters $q = 3$ and $\gamma = 0.5$. However, we observed that by choosing convenient combinations of $q \in \{3, 5, 10\}$, $\gamma = \{0.3, 0.5\}$ and $\beta = \{10, 100\}$ the globalized Newton strategy converged to a solution of (1) for all the 24 problems of this subsection.

Table 7: Comparison between monotone (TR) and tolerant strategies.

| Problem | TR | Newton | Broyden | Column–Updating |
|---|---|---|---|---|
| D1, $n = 50$ | 64, 127, 3.E-7 | 6, 13, 6.E-12 (2) | 7, 14, 6.E-10 (2) | 7, 14, 6.E-10 (2) |
| D1, $n = 100$ | 65, 129, 4.E-7 | 6, 13, 2.E-12 (2) | 7, 14, 2.E-10 (2) | 7, 14, 2.E-10 (2) |
| D2, $n = 4$ | 13, 25, 4.E-10 | 5, 11, 3.E-8 (2) | 7, 13, 9.E-8 (2) | 7, 13, 6.E-15 (2) |
| D2, $n = 52$ | 13, 25, 8.E-15 | 5, 13, 9.E-10 (2) | 7, 15, 4.E-9 (2) | 7, 15, 4.E-18 (2) |
| D2, $n = 100$ | 13, 24, 1.E-14 | 6, 14, 1.E-17 (2) | 8, 16, 5.E-13 (2) | 7, 15, 1.E-10 (2) |
| D3, $n = 2$ | 100, 201, 1.E-4 | 6, 9, 1.E-4 (3) | 6, 9, 1.E-4 (3) | 6, 9, 1.E-4 (3) |
| D3, $n = 50$ | 100, 201, 1.E-4 | 6, 9, 5.E-4 (3) | 6, 9, 5.E-4 (3) | 6, 9, 5.E-4 (3) |
| D4, $n = 3$ | 64, 125, 9.E-5 | 15, 18, 2.E-11 (10) | 18, 21, 2.E-10 (10) | 18, 21, 7.E-10 (10) |
| D4, $n = 51$ | 84, 165, 9.E-5 | 14, 19, 7.E-11 (10) | 16, 21, 3.E-10 (10) | 16, 21, 2.E-10 (10) |
| D4, $n = 99$ | 87, 171, 9.E-5 | 17, 22, 1.E-14 (10) | 17, 22, 2.E-9 (10) | 17, 22, 2.E-9 (10) |
| D5, $n = 33$ | 22, 55, 5.E-6 | 6, 9, 7.E-12 (3) | 6, 9, 6.E-8 (3) | 6, 9, 6.E-8 (3) |
| D5, $n = 78$ | 22, 55, 5.E-6 | 7, 10, 4.E-15 (2) | 9, 12, 5.E-9 (2) | 9, 12, 7.E-10 (2) |
| D5, $n = 99$ | 5, 7, 3.E-11 | 7, 10, 5.E-15 (1) | 12, 15, 2.E-9 (1) | 12, 15, 2.E-9 (1) |
| D6, $n = 10$ | 19, 34, 1.8 | 8, 9, 6.E-9 (0) | 17, 19, 4.E-8 (1) | 17, 19, 6.E-8 (1) |
| D6, $n = 50$ | 20, 45, 0.27 | 11, 12, 7.E-12 (0) | 21, 24, 3.E-11 (2) | 19, 21, 3.E-8 (1) |
| D6, $n = 100$ | 24, 76, 1.9 | 11, 12, 9.E-8 (0) | 21, 23, 1.E-8 (1) | 22, 24, 1.E-8 (1) |
| D7, $n = 3$ | 27, 56, 0.66 | 15, 34, 2.E-14 (5) | 21, 273, 0.66 (14) | 13, 18, 1.E-9 (2) |
| D7, $n = 33$ | 37, 76, 2.2 | 23, 306, 2.2 (16) | 15, 25, 9.E-12 (4) | 19, 233, 2.2 (12) |
| D7, $n = 99$ | 39, 80, 3.8 | 17, 218, 3.8 (10) | 17, 181, 3.8 (11) | 24, 221, 3.8 (20) |
| D8, $n = 3$ | 26, 51, 0.1 | 10, 13, 3.E-14 (2) | 100, 296, 8.E-2 (93) | 100, 296, 8.E-2 (93) |
| D8, $n = 33$ | 34, 67, 0.3 | 10, 14, 2.E-11 (2) | 100, 300, 0.3 (93) | 100, 297, 0.3 (93) |
| D8, $n = 99$ | 36, 71, 0.6 | 10, 14, 4.E-11 (2) | 100, 302, 0.5 (93) | 100, 298, 0.5 (93) |
| D9, $n = 10$ | 39, 76, 1.59 | 71, 181, 0.5 (28) | 85, 168, 9.E-12 (31) | 57, 147, 4.E-11 (17) |
| D10 | 5, 7, 3.E-7 | 7, 20, 4.E-9 (2) | 13, 26, 1.E-8 (2) | 14, 26, 6.E-9 (2) |

In terms of overall computational effort, the total number of TR iterations was 958 with 1949 functional evaluations. Newton's tolerant strategy performed 299 iterations, 119 of them special and evaluated the function 1003 times. The figures for the quasi-Newton tolerant strategies are 636 iterations of Broyden (397 special) with 1817 functional evaluations and 610 iterations for Column-Updating (387 special) with 1789 functional evaluations. However, considering only the 12 problems that succeeded for all the strategies, we had 915 functional evaluations for TR and 172, 198 and 197 for Newton, Broyden and Column-Updating, respectively. These results seem to reinforce the belief that being tolerant is worthwhile.

# 6 FINAL REMARKS

The size and shape of the convergence basin of a local method for solving nonlinear systems depend strongly on structural characteristics of the system as well as on the level sets of $\|F(x)\|$. Points where $\|F(x)\|$ is small are more likely to belong to such a convergence region. However, the norm of $F$ is not invariant under linear changes on the range space of $F$, while many local methods are. Therefore, purely monotone strategies can reject iterates belonging to rapidly convergent sequences, even if they are relatively close to the solution. Moreover, strictly monotone strategies usually have local-nonglobal minimizers of $\|F(x)\|$ as attraction points, which are completely useless when our aim is to solve $F(x) = 0$. The attraction of local minimizers and the possibility of frequent rejection of good iterates are our main motivations for the development of nonmonotone strategies. In this paper we exploited a very simple idea for combining global and local iterations. In spite of its simplicity, the implemented strategy is able to take into account in an efficient way the characteristics of quasi-Newton methods that do not necessarily generate descent directions for $\|F(x)\|$.

Both Broyden's method and the Column-Updating method are affine invariant in the range space, as it can be easily verified from equation (8). Of course, the same property holds for Newton's method. This means that the iterations of these methods applied to $AF(x) = 0$ are the same as the iterations applied to $F(x) = 0$, provided that $A$ is nonsingular and that the first Jacobian approximation is chosen conveniently (for example, when $B_0$ is the Jacobian $J(x^0)$). Some authors detected that affine invariancy is an important feature that is related with practical efficiency and developed globalization strategies that try to keep this property. See [11] and references therein. The tolerant strategy is also related to the necessity of not destroying affine invariancy: local (affine invariant) iterations are executed whenever it is possible and, in fact, they are abandoned only when the property fails to prove its utility. Active research is taking place with the aim of incorporating affine invariancy properties in the inexact Newton approach. See [57], [13], [43], [44], [46]. This research is related to the possibility of preconditioning efficiently the underlying linear problem.

For particular nonlinear systems other objective functions (different from the squared norm of $F$) can be useful. The ideal situation is when the stationary points of the objective function coincide with the solutions of (1). Friedlander, Martinez and Santos [19], [17], [20] introduced functions with this property for nonlinear systems related to linearly constrained optimization, complementarity problems and variational inequalities. In these problems, only positive solutions ($x \geq 0$) are meaningful, so, we can take advantage of the possibility of modifying the special iterations to deal with box constraints.

## References


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

[2] Brown, P.N. and Saad, Y. (1990). Hybrid Krylov methods for nonlinear systems of equations, *SIAM Journal on Scientific and Statistical Computing* **11**, 450-481.

[3] Brown, P.N. and Saad, Y. (1994). Convergence theory of nonlinear Newton-Krylov methods, *SIAM Journal on Optimization* **4**, 297-330.

[4] Broyden, C.G., Dennis Jr., J.E. and Moré, J.J. (1973). On the local and superlinear convergence of quasi-Newton methods, *Journal of the Institute of Mathematics and its Applications* **12**, 223-245.

[5] Conn, A.R., Gould, N.I.M. and Toint, Ph.L. (1989). Testing a class of methods for solving minimization problems with simple bounds on the variables, *Mathematics of Computation* **50**, 399-430.

[6] Dembo, R.S., Eisenstat, S.C. and Steihaug, T. (1982). Inexact Newton methods, *SIAM Journal on Numerical Analysis* **19**, 400-408.

[7] Dennis Jr., J.E., Martinez, J.M. and Zhang, X. (1994). Triangular decomposition methods for solving reducible nonlinear systems of equations, *SIAM Journal on Optimization* **4**, 358-382.

[8] Dennis Jr., J.E. and Schnabel, R.B. (1979). Least change secant updates for quasi-Newton methods, *SIAM Review* **21**, 443-459.

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

[10] Dennis Jr., J.E. and Walker, H.F. (1981). Convergence theorems for least-change secant update methods, *SIAM Journal on Numerical Analysis* **18**, 949-987.

[11] Deuflhard, P. (1991). Global inexact Newton methods for very large scale nonlinear problems, *Impact of Computing in Science and Engineering* **3**, 366-393.

[12] Deuflhard, P. (1995). *Newton techniques for highly nonlinear problems. Theory and algorithms*, in preparation.

[13] Deuflhard, P., Freund, R. and Walter, A. (1990). Fast secant methods for the iterative solution of large nonsymmetric linear systems, *Impact of Computing in Science and Engineering* **2**, 244-276.

[14] Diniz–Ehrhardt, M.A. and Martinez, J.M. (1993). A parallel projection method for overdetermined nonlinear systems of equations, *Numerical Algorithms* **4**, 241-262.

[15] Eisenstat, S.C. and Walker, H.F. (1994). Globally convergent inexact Newton methods, *SIAM Journal on Optimization* **4**, 393-422.

[16] Friedlander, A. and Martinez, J.M. (1994). On the maximization of a concave quadratic function with box constraints, *SIAM Journal on Optimization.* **4**, 177-192.

[17] Friedlander, A., Martinez, J.M. and Santos, S.A. (1995). Resolution of linear complementarity problems using minimization with simple bounds, *Journal of Global Optimization* **6**, 1-15.

[18] Friedlander, A., Martinez, J.M. and Santos, S.A. (1994). A new trust-region algorithm for bound constrained minimization, *Applied Mathematics and Optimization* **30**, 235-266.

[19] Friedlander, A., Martinez, J.M. and Santos, S.A. (1994). On the resolution of linearly constrained convex minimization problems, *SIAM Journal on Optimization* **4**, 331-339.

[20] Friedlander, A., Martinez, J.M. and Santos, S.A. (1995). Solving variational inequality problems on bounded polytopes, *Numerical Functional Analysis and Optimization* **16**, 653-668.

[21] Gay, D.M. (1979). Some convergence properties of Broyden's method, *SIAM Journal on Numerical Analysis* **16**, 623-630.

[22] George, A. and Ng, E. (1987). Symbolic factorization for sparse Gaussian elimination with partial pivoting, *SIAM Journal on Scientific and Statistical Computing* **8**, 877-898.

[23] Golub, G.H. and Van Loan, Ch.F. (1989). *Matrix Computations*, The Johns Hopkins University Press, Baltimore and London.

[24] Gomes–Ruggiero, M.A. and Martinez, J.M. (1992). The Column–Updating Method for solving nonlinear equations in Hilbert space, *RAIRO Mathematical Modelling and Numerical Analysis* **26**, 309-330.

[25] Gomes–Ruggiero, M.A., Martinez, J.M. and Moretti, A.C. (1992). Comparing algorithms for solving sparse nonlinear systems of equations, *SIAM Journal on Scientific and Statistical Computing* **13**, 459-483.

[26] Griewank, A. (1992). Achieving Logarithmic Growth of Temporal and Spacial Complexity in Reverse Automatic Differentiation, *Optimization Methods and Software* **1**, 35-54.

[27] Griewank, A. and Toint, Ph.L. (1982). On the unconstrained optimization of partially separable functions, in *Nonlinear Optimization 1981*, edited by M.J.D. Powell, Academic Press, New York.

[28] Griewank, A. and Toint, Ph.L. (1982). Partitioned variable metric for large structured optimization problems, *Numerische Mathematik* **39**, 119-137.

[29] Griewank, A. and Toint, Ph.L. (1982). Local convergence analysis for partitioned quasi-Newton updates, *Numerische Mathematik* **39**, 429-448.

[30] Griewank, A. and Toint, Ph.L. (1984). Numerical experiments with partially separable optimization problems, in *Numerical Analysis Proceedings Dundee 1983*, edited by D.F. Griffiths, Lecture Notes in Mathematics vol. 1066, Springer - Verlag, Berlin, 203-220.

[31] Grippo, L., Lampariello, F. and Lucidi, S. (1986). A nonmonotone line search technique for Newton's method, *SIAM Journal on Numerical Analysis* **23**, 707-716.

[32] Grippo, L., Lampariello, F. and Lucidi, S. (1991). A class of nonmonotone stabilization methods in unconstrained optimization, *Numerische Mathematik* **59**, 779-805.

[33] Hestenes, M.R. and Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems, *Journal of Research of the National Bureau of Standards* **B49**, 409-436.

[34] Johnson, G.W. and Austria, N.H. (1983). A quasi-Newton method employing direct secant updates of matrix factorizations, *SIAM Journal on Numerical Analysis* **20**, 315-325.

[35] Kelley, C.T. (1995). *Iterative methods for linear and nonlinear equations*, SIAM Publications, Philadelphia.

[36] Kelley, C.T. and Sachs, E.W. (1987). A quasi-Newton method for elliptic boundary value problems, *SIAM Journal on Numerical Analysis* **24**, 516-531.

[37] Lukšan, L. (1994). Inexact trust region method for large sparse systems of nonlinear equations, *Journal of Optimization Theory and Applications* **81**, 569-590.

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

[39] Martinez, J.M. (1990). A family of quasi-Newton methods for nonlinear equations with direct secant updates of matrix factorizations, *SIAM Journal on Numerical Analysis* **27**, 1034-1049.

[40] Martinez, J.M. (1990). Local convergence theory of inexact Newton methods based on structured least change updates, *Mathematics of Computation* **55**, 143-168.

[41] Martinez, J.M. (1992). On the relation between two local convergence theories of least change secant update methods, *Mathematics of Computation* **59**, 457-481.

[42] Martinez, J.M. (1992). Fixed-Point Quasi-Newton Methods, *SIAM Journal on Numerical Analysis* **29**, 1413-1434.

[43] Martinez, J.M. (1993). A Theory of Secant Preconditioners, *Mathematics of Computation* **60**, 681-698.

[44] Martinez, J.M. (1995). An extension of the theory of secant preconditioners, *Journal of Computational and Applied Mathematics* **60**, 115-125.

[45] Martinez, J.M. (1993). On the Convergence of the Column-Updating Method, *Matemática Aplicada e Computacional* **12**, 83-94.

[46] Martinez, J.M. and Qi, L. (1995). Inexact Newton methods for solving nonsmooth equations, *Journal of Computational and Applied Mathematics* **60**, 127-145.

[47] Moré, J.J. and Cosnard, M.Y. (1979). Numerical comparison of three nonlinear equation solvers, *ACM Transactions on Mathematical Software* **5**, 64-85.

[48] Moré, J.J. and Sorensen, D.C. (1983). Computing a trust region step, *SIAM Journal on Scientific and Statistical Computing* **4**, 553-572.

[49] Ortega, J.M. and Rheinboldt, W.G. (1970).*Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, NY.

[50] Raydan, M. (1994). The Barzilai-Borwein gradient method for the large scale unconstrained minimization problem, Technical Report 94-01, Departamento de Matemáticas, Universidad Central de Venezuela, to appear in *SIAM Journal on Optimization.*

[51] Saad, Y. and Schultz, M.H. (1986). GMRES. A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM Journal on Numerical Analysis* **7**, 856-869.

[52] Schreiber R. and Keller, H.B. (1983). Driven cavity flows by efficient numerical techniques, *Journal of Computational Physics* **49**, 310-333.

[53] Spedicato, E. (1975). Computational experience with quasi-Newton algorithms for minimization problems of moderately large size, Report CISE-N-175, Segrate (Milano).

[54] Toint, Ph.L. (1986). Numerical solution of large sets of algebraic non-linear equations, *Mathematics of Computation* **16**, 175-189.

[55] Toint, Ph. L. (1995). An assessment of non-monotone linesearch techniques for unconstrained optimization, *SIAM Journal on Scientific Computing*, to appear.

[56] Walker, H.F. (1992). A GMRES-backtracking Newton iterative method, *Proceedings of the Copper Mountain Conference on Iterative Methods, April 1992.*

[57] Ypma, T.J. (1984). Local convergence of inexact Newton methods, *SIAM Journal on Numerical Analysis* **21**, 583-590.