Spectral residual method without gradient information for solving large-scale nonlinear systems of equations: Theory and experiments

William La Cruz
Departamento de Electrónica, Computación y Control
Universidad Central de Venezuela
Caracas, Venezuela.
E-mail: wlacruz@elecrisc.ing.ucv.ve

José Mario Martínez †
Department of Applied Mathematics IMECC-UNICAMP
University of Campinas
CP 6065, 13081-970 Campinas SP, Brazil
E-mail: martinez@ime.unicamp.br

Marcos Raydan‡
Departamento de Computación
Universidad Central de Venezuela
Apartado 47002, Caracas 1041-A, Venezuela
E-mail: mraydan@reacciun.ve

Technical Report RT-04-08, July 2004

Abstract
A fully derivative-free spectral residual method for solving large-scale nonlinear systems of equations is presented. It uses in a systematic way the residual vector as a search direction, a spectral step length that produces a nonmonotone process and a globalization strategy that allows this nonmonotone behavior. The global convergence analysis of the combined scheme is presented. An extensive set of numerical experiments that indicate that the new combination is competitive and frequently better than well-known Newton Krylov methods for large-scale problems is also presented.

*Supported by Agenda Petróleo UCV-PROJECT 97-003769.
†Supported by PRONEX-Optimization 76.79.1008-00, FAPESP (Grant 90-3724-6).
‡Supported by Agenda Petróleo UCV-PROJECT 97-003769.
1 Introduction

We introduce a derivative-free nonmonotone iterative method for solving the nonlinear system of equations

\[ F(x) = 0, \]  

where \( F : \mathbb{R}^n \to \mathbb{R}^n \) is a continuously differentiable mapping. We are interested in large-scale systems for which the Jacobian of \( F \) is not available or requires a prohibitive amount of storage.

Recently, La Cruz and Raydan [22] introduced the Spectral Algorithm for Nonlinear Equations (SANE) for solving (1). SANE uses in a systematic way the residual \( \pm F(x_k) \) as search direction. The first trial point at each iteration is \( x_k - \sigma_k F(x_k) \) where \( \sigma_k \) is a spectral coefficient. Global convergence is guaranteed by means of a variation of the nonmonotone strategy of Grippo, Lampariello and Lucidi [19]. This approach requires descent directions with respect to the squared norm of the residual. As a consequence, the computation of a directional derivative, or a very good approximation of it, is necessary at every iteration.

The spectral coefficient is an appropriate Rayleigh quotient with respect to a secant approximation of the Jacobian. Spectral gradient methods for minimization were originated in the Barzilai-Borwein paper [2]. The properties of their method for general quadratic functions were elucidated in [30]. Further analysis of spectral gradient methods can be found in [10, 14, 31] among others. For a review containing the more recent advances on spectral choices of the steplength for minimization problems, see [15].

In this paper we introduce a new nonmonotone line-search technique that can be associated to the same search directions and the same initial steplengths as the SANE algorithm. In other words, the first trial point at each iteration will be the same as in SANE, but the line-search strategy is different. The main consequence is that, in the new approach, directional derivatives are not required at all.

We also present an extensive set of numerical experiments that indicate that the new method is competitive and sometimes better than the SANE algorithm. We recall that, in [22], SANE was in turn compared favorably with several Newton-Krylov methods (see, e.g., [3, 8, 9]). Therefore, the new algorithm represents an encouraging low-cost scheme for solving (1).

Notation.

- \( J(x) \) will denote the Jacobian matrix of \( F \) computed at \( x \).
- For all \( x \in \mathbb{R}^n \) we denote \( g(x) = 2J(x)^t F(x) = \nabla \| F(x) \|_2^2 \).
- The set of natural numbers will be denoted \( \mathbb{N} = \{0, 1, 2, \ldots\} \).
- If \( \{z_k\}_{k \in \mathbb{N}} \) is a sequence and \( K = \{k_1, k_2, k_3, \ldots\} \) is an infinite subset of \( \mathbb{N} \) such that \( k_i < k_j \) if \( i < j \), we denote:

\[ \lim_{k \in K} z_k = \lim_{j \to \infty} z_{k_j}. \]
- The symbol \( \| \cdot \| \) will always denote the Euclidian norm.
2 The new nonmonotone line-search strategy

The best known nonmonotone line-search technique for unconstrained optimization was introduced by Grippo, Lampariello and Lucidi [19]. It has been used to globalize the spectral gradient method [31] and some of its extensions for convex constrained optimization [5, 6] and nonlinear systems of equations [22]. Different nonmonotone line-search techniques, associated to Newton and quasi-Newton strategies, have been proposed for solving (1). See [17, 24]. Li and Fukushima [24] presented an interesting idea by means of which descent directions are not necessary to guarantee that each iteration is well defined. Let us briefly describe the Grippo-Lampariello-Lucidi (GLL) and the Li-Fukushima (LF) schemes.

The GLL condition can be written as follows:

\[ f(x_k + \alpha_k d_k) \leq \max_{0 \leq j \leq M-1} f(x_{k-j}) + \gamma \alpha_k \nabla f(x_k)^t d_k, \]

where \( M \) is a nonnegative integer, \( 0 < \gamma < 1 \) and \( f \) is a merit function such that \( f(x) = 0 \) if and only if \( \| F(x) \| = 0 \).

The LF condition can be written as follows:

\[ \| F(x_k + \alpha_k d_k) \| \leq (1 + \eta_k) \| F(x_k) \| - \gamma \alpha_k^2 \| d_k \|^2, \]

where \( \sum_k \eta_k \leq \eta < \infty \).

It follows that if \( \nabla f(x_k)^t d_k < 0 \), then the GLL condition is satisfied for \( \alpha_k \) sufficiently close to zero and we can compute a steplength \( \alpha_k \) by using a finite backtracking process. However, when \( \nabla f(x_k)^t d_k = 0 \), the existence of \( \alpha_k \) satisfying the GLL condition is not guaranteed. Unfortunately, when \( d_k = \pm F(x_k) \) for solving (1), \( \nabla f(x_k)^t d_k = \pm 2F(x_k)^t J(x_k) F(x_k) \) could be close to zero or zero, and then stagnation or breakdown will occur during the backtracking process.

One possible remedy is to use the LF condition. This condition does not need the computation of \( J(x_k)d_k \). Moreover, it is satisfied, if \( \alpha_k \) is small enough, independently of the choice of \( d_k \). However, since \( \eta_k \) is usually very small when \( k \) is large, the Li-Fukushima strategy generally imposes an almost monotone behavior of the merit function when \( x_k \) is close to a solution. This is not a good feature when one uses spectral gradient or spectral residual steps because, in these cases, the pure undamped methods (where \( \alpha_k = 1 \) for all \( k \)), although generally effective, use to be highly nonmonotone even in the neighborhood of an isolated solution. The reason for this is not completely understood but the analogy with the behavior of the spectral gradient (or Barzilai-Borwein) method for minimizing convex quadratics may be useful. See [30]. In the quadratic case the spectral gradient method does not need line-search strategies for being globally convergent, but the functional values do not decrease monotonically at all. Therefore, imposing any kind of monotonicity is not convenient. Many authors, including Fletcher [15], pointed out the necessity of avoiding monotonicity requirements in the spectral framework as much as possible.
In this work we combine and extend the GLL and LF conditions to produce a robust nonmonotone line-search globalization strategy that somehow takes into account the advantages of both schemes. Roughly speaking the new descent condition can be written as:

\[ f(x_{k+1}) \leq \max_{0 \leq j \leq M-1} f(x_{k-j}) + \eta_k - \gamma \alpha_k^2 f(x_k). \quad (2) \]

The GLL term \( \max_{0 \leq j \leq M-1} f(x_{k-j}) \) is responsible for the sufficiently nonmonotone behavior of \( f(x_k) \) even when \( k \) is large. On the other hand, the presence of \( \eta_k > 0 \) guarantees that all the iterations are well defined and the forcing term \( -\gamma \alpha_k^2 f(x_k) \) provides the arguments for proving global convergence.

We would like to mention that a similar extension that also combines the GLL with the LF conditions was presented and briefly discussed in the final remarks of [22]. However, it was neither analyzed nor tested. The present work was motivated by the need of studying the theoretical and practical properties of this globalization technique.

3 Model algorithm and convergence

We assume that \( F : \mathbb{R}^n \rightarrow \mathbb{R}^n \) has continuous partial derivatives. Let \( nexp \in \{1, 2\} \). Define

\[ f(x) = \|F(x)\|^{nexp} \quad \text{for all} \quad x \in \mathbb{R}^n. \]

Assume that \( \{\eta_k\} \) is a sequence such that \( \eta_k > 0 \) for all \( k \in \mathbb{N} \) and

\[ \sum_{k=0}^{\infty} \eta_k = \eta < \infty. \quad (3) \]

Assume that \( 0 < \gamma < 1 \) and \( 0 < \sigma_{\min} < \sigma_{\max} < \infty \). Let \( M \) be a positive integer. Let \( \tau_{\min}, \tau_{\max} \) be such that \( 0 < \tau_{\min} < \tau_{\max} < 1 \).

Given \( x_0 \in \mathbb{R}^n \) an arbitrary initial point, the algorithm that allows us to obtain \( x_{k+1} \) starting from \( x_k \) is given below.

Algorithm DF-SANE (Derivative free SANE).

Step 1.

- Choose \( \sigma_k \) such that \( |\sigma_k| \in [\sigma_{\min}, \sigma_{\max}] \) (the spectral coefficient)
- Compute \( \bar{f}_k = \max\{f(x_k), \ldots, f(x_{\max\{0,k-M+1\}})\} \).
- Set \( d \leftarrow -\sigma_k F(x_k) \).
- Set \( \alpha_+ \leftarrow 1, \alpha_- \leftarrow 1. \)

Step 2.

If \( f(x_k + \alpha_+ d) \leq \bar{f}_k + \eta_k - \gamma \alpha_+^2 f(x_k) \) then

Define \( d_k = d, \alpha_k = \alpha_+, x_{k+1} = x_k + \alpha_k d_k \)

else if \( f(x_k - \alpha_- d) \leq \bar{f}_k + \eta_k - \gamma \alpha_-^2 f(x_k) \) then

Define \( d_k = -d, \alpha_k = \alpha_-, x_{k+1} = x_k + \alpha_k d_k \)
else
  choose $\alpha_{+\text{new}} \in [\tau_{\min}\alpha_+, \tau_{\max}\alpha_+]$, $\alpha_{-\text{new}} \in [\tau_{\min}\alpha_-, \tau_{\max}\alpha_-]$,
  replace $\alpha_+ \leftarrow \alpha_{+\text{new}}$, $\alpha_- \leftarrow \alpha_{-\text{new}}$
  and go to Step 2.

**Remark.** As we will see later, the coefficient $\sigma_k$ will be intended to be an approximation
of the quotient $\|F(x_k)\|^2 / \langle J(x_k)F(x_k), F(x_k) \rangle$. This quotient may be positive or negative (or even null).

**Proposition 1.** The iteration is well defined.

**Proof.** Since $\eta_k > 0$, after a finite number of reductions of $\alpha_+$ the condition

$$f(x_k + \alpha_+d) \leq \bar{f}_k + \eta_k - \gamma \alpha_+^2 f(x_k)$$

necessarily holds. \qed

In order to prove the convergence of the algorithm, we need some preliminary definitions (see [7]). For all $k = 1, 2, \ldots$ we define:

$$V_k = \max\{f(x_{(k-1)M+1}), \ldots, f(x_{kM})\}.$$

Let $\nu(k) \in \{(k-1)M + 1, \ldots, kM\}$ be such that, for all $k = 1, 2, \ldots$,

$$f(x_{\nu(k)}) = V_k.$$

Clearly,

$$f(x_{kM+1}) \leq \max\{f(x_{(k-1)M+1}), \ldots, f(x_{kM})\} + \eta_{kM} - \gamma \alpha_{kM}^2 f(x_{kM})$$

$$= V_k + \eta_{kM} - \gamma \alpha_{kM}^2 f(x_{kM})$$

$$\leq V_k + \eta_{kM},$$

$$f(x_{kM+2}) \leq \max\{V_k, f(x_{kM+1})\} + \eta_{kM+1} - \gamma \alpha_{kM+1}^2 f(x_{kM+1})$$

$$\leq V_k + \eta_{kM} + \eta_{kM+1} - \gamma \alpha_{kM+1}^2 f(x_{kM+1})$$

$$\leq V_k + \eta_{kM} + \eta_{kM+1}$$

and so on.

Therefore, by an inductive argument,

$$f(x_{kM+\ell}) \leq V_k + \sum_{j=0}^{\ell-1} \eta_{kM+j} - \gamma \alpha_{kM+\ell-1}^2 f(x_{kM+\ell-1}) \quad (4)$$

for all $\ell = 1, 2, \ldots, M$.

But $\nu(k+1) \in \{kM + 1, \ldots, kM + M\}$, thus,

$$V_{k+1} = f(x_{\nu(k+1)}) \leq V_k + \sum_{j=0}^{M-1} \eta_{kM+j} - \gamma \alpha_{\nu(k+1)-1}^2 f(x_{\nu(k+1)-1})$$

5
So, for all \( k = 1, 2, \ldots \) we have that
\[
 f(x_{\nu(k+1)}) \leq f(x_{\nu(k)}) + \sum_{j=0}^{M-1} \eta_{kM+j} - \gamma \alpha_{\nu(k+1)-1}^2 f(x_{\nu(k+1)-1}).
\] (5)

Using (4) and (5) we can prove the following propositions:

**Proposition 2.** For all \( k, \ell = 1, 2, \ldots \),
\[
f(x_{kM+\ell}) \leq f(x_{\nu(k)}) + \sum_{i=\nu(k)}^{\infty} \eta_i \leq f(x_{\nu(k)}) + \eta.
\] (6)

**Proof.** Straightforward, using (4) and (5). \(\square\)

**Proposition 3.**
\[
\lim_{k \to \infty} \alpha_{\nu(k)-1}^2 f(x_{\nu(k)-1}) = 0.
\]

**Proof.** Write the inequalities (5) for \( k = 1, 2, \ldots, L \). Observe that \( f(x_{\nu(k+1)}) \) occurs on the left-hand side of the \( k \)-th inequality and also on the right-hand side of the \( k+1 \)-th inequality. Adding the \( L \) inequalities, we get:
\[
f(x_{\nu(L+1)}) \leq f(x_{\nu(1)}) + \sum_{j=M}^{(L+1)M-1} \eta_j - \gamma \sum_{j=1}^{L} \alpha_{\nu(j+1)-1}^2 f(x_{\nu(j+1)-1}).
\]

Therefore, for all \( L = 1, 2, \ldots \), we obtain:
\[
\gamma \sum_{j=1}^{L} \alpha_{\nu(j+1)-1}^2 f(x_{\nu(j+1)-1}) \leq f(x_{\nu(1)}) + \sum_{j=M}^{(L+1)M-1} \eta_j - f(x_{\nu(L+1)}) \leq f(x_{\nu(1)}) + \sum_{j=M}^{(L+1)M-1} \eta_j \leq f(x_{\nu(1)}) + \eta
\]

So, the series \( \sum_{j=1}^{\infty} \alpha_{\nu(j+1)-1}^2 f(x_{\nu(j+1)-1}) \) is convergent. This implies the desired result. \(\square\)

From now on we define:
\[
K = \{\nu(1) - 1, \nu(2) - 2, \nu(3) - 3, \ldots\}. \quad (7)
\]

and
\[
K_+ = \{\nu(1), \nu(2), \nu(3), \ldots\}. \quad (8)
\]

Observe that
\[
\nu(j+1) \leq \nu(j) + 2M - 1 \text{ for all } j = 1, 2, \ldots \quad (9)
\]

In Theorem 1 we prove that, at every limit point \( x_\ast \) of the subsequence \( \{x_k\}_{k \in K} \) one necessarily has that \( \langle J(x_\ast)F(x_\ast), F(x_\ast) \rangle = \langle F(x_\ast), g(x_\ast) \rangle = 0 \). In other words the gradient of \( \|F(x)\|^2 \) at \( x_\ast \) is orthogonal to the residual \( F(x_\ast) \).
Theorem 1. Assume that \( \{x_k\}_{k \in \mathbb{N}} \) is generated by the DF-SANE algorithm. Then, every limit point \( x_* \) of \( \{x_k\}_{k \in K} \) satisfies
\[
\langle F(x_*), J(x_*)^T F(x_*) \rangle = 0. \tag{10}
\]

Proof. By Proposition 3 we have that
\[
\lim_{k \in K} \alpha^2_k f(x_k) = 0. \tag{11}
\]
Let \( x_* \) be a limit point of \( \{x_k\}_{k \in K} \). Let \( K_1 \subset K \) be an infinite sequence of indices such that
\[
\lim_{k \in K_1} x_k = x_*.
\]
Then, by (11),
\[
\lim_{k \in K_1} \alpha^2_k f(x_k) = 0. \tag{12}
\]
If \( \{\alpha_k\}_{k \in K_1} \) does not tend to zero, there exists an infinite sequence of indices \( K_2 \subset K_1 \) such that \( \alpha_k \) is bounded away from zero for \( k \in K_2 \). Then, by (12),
\[
\lim_{k \in K_2} f(x_k) = 0.
\]
Since \( f \) is continuous and \( \lim_{k \in K_2} x_k = x_* \) this implies that \( f(x_*) = 0 \).
So, we only need to analyze the case
\[
\lim_{k \in K_1} \alpha_k = 0. \tag{13}
\]
At Step 2 of Algorithm DF-SANE one tests the inequality
\[
f(x_k + \alpha_k^+ d) \leq \bar{f}_k + \eta_k - \gamma \alpha_k^2 f(x_k). \tag{14}
\]
If (14) does not hold, the inequality
\[
f(x_k - \alpha_k^- d) \leq \bar{f}_k + \eta_k - \gamma \alpha_k^2 f(x_k) \tag{15}
\]
is tested.

The first trial steps at (14)-(15) are \( \alpha_+ = \alpha_- = 1 \). By (13), there exists \( k_0 \in K_1 \) such that \( \alpha_k < 1 \) for all \( k \geq k_0, k \in K_1 \). Therefore, for those iterations \( k \), the step \( \alpha_k \) that fulfills (14) or (15) is necessarily preceded by steps \( \alpha_k^+ \) and \( \alpha_k^- \) that do not satisfy (14) and (15) respectively. Now, by the choice of \( \alpha_{+\text{new}} \) and \( \alpha_{-\text{new}} \) at Step 2 of the algorithm, for all \( k \geq k_0, k \in K_1 \), there exists \( m_k \in \mathbb{N} \) such that
\[
\alpha_k \geq \tau^{m_k}_{\min}.
\]
So, by (13),
\[
\lim_{k \in K_1} m_k = \infty.
\]
But, again by the choice of \( \alpha_{+\text{new}} \) and \( \alpha_{-\text{new}} \),
\[
\alpha_k^+ \leq \tau^{m_k-1}_{\max}.
\]
and 
\[ \alpha_k^- \leq \tau_{\text{max}}^{m_k-1}. \]

Therefore, since \( \tau_{\text{max}} < 1 \),
\[ \lim_{k \to K_1} \alpha_k^+ = \lim_{k \to K_1} \alpha_k^- = 0. \]

Clearly, the fact that (14) and (15) are not satisfied by \( \alpha_k^+ \) and \( \alpha_k^- \) respectively implies that
\[ f(x_k - \alpha_k^+ \sigma_k F(x_k)) > \bar{f}_k + \eta_k - \gamma (\alpha_k^+) \]
and
\[ f(x_k + \alpha_k^- \sigma_k F(x_k)) > \bar{f}_k + \eta_k - \gamma (\alpha_k^-) \]
for all \( k \in K_1, k \geq k_0 \).

The inequality (16) implies that
\[ f(x_k - \alpha_k^+ \sigma_k F(x_k)) > f(x_k) - \gamma (\alpha_k^+)^2 f(x_k). \]

So,
\[ f(x_k - \alpha_k^+ \sigma_k F(x_k)) - f(x_k) \geq -\gamma (\alpha_k^+)^2 f(x_k). \]

By Proposition 2, \( \{f(x_k)\}_{k \in \mathcal{K}} \) is bounded above, say, by a constant \( c > 0 \). Thus,
\[ f(x_k - \alpha_k^+ \sigma_k F(x_k)) - f(x_k) \geq -c\gamma (\alpha_k^+)^2. \]

If \( n_{\text{exp}} = 2 \) this means that
\[ \|F(x_k - \alpha_k^+ \sigma_k F(x_k))\|^2 - \|F(x_k)\|^2 \geq -c\gamma (\alpha_k^+)^2. \]

Now, the subsequence \( \{x_k\}_{k \in K_1} \) is convergent and, therefore, bounded. Since \( \|F(x_k)\|, \alpha_k^+ \) and \( \sigma_k \) are also bounded, we have that \( \{x_k - \alpha_k^+ \sigma_k F(x_k)\}_{k \in K_1} \) is bounded. So, by the continuity of \( F \), there exists \( c_1 > 0 \) such that
\[ \|F(x_k - \alpha_k^+ \sigma_k F(x_k))\| + \|F(x_k)\| \leq c_1 \text{ for all } k \in K_1. \]

So, if \( n_{\text{exp}} = 1 \), multiplying both sides of (18) by \( \|F(x_k - \alpha_k^+ \sigma_k F(x_k))\| + \|F(x_k)\| \), we obtain:
\[ \|F(x_k - \alpha_k^+ \sigma_k F(x_k))\|^2 - \|F(x_k)\|^2 \geq -cc_1\gamma (\alpha_k^+)^2. \]

Thus, taking \( C = \max\{c, cc_1\} \) we obtain, for \( n_{\text{exp}} \in \{1, 2\} \), that
\[ \|F(x_k - \alpha_k^+ \sigma_k F(x_k))\|^2 - \|F(x_k)\|^2 \geq -C\gamma (\alpha_k^+)^2. \]

So,
\[ \frac{\|F(x_k - \alpha_k^+ \sigma_k F(x_k))\|^2 - \|F(x_k)\|^2}{\alpha_k^+} \geq -C\gamma \alpha_k^+. \]

By the Mean Value Theorem, there exists \( \xi_k \in [0, 1] \) such that
\[ \langle g(x_k - \xi_k \alpha_k^+ \sigma_k F(x_k)), -\sigma_k F(x_k) \rangle \geq -C\gamma \alpha_k^+. \]

Therefore,
\[ \sigma_k \langle g(x_k - \xi_k \alpha_k^+ \sigma_k F(x_k)), -F(x_k) \rangle \geq -C\gamma \alpha_k^+. \]

(19)
Now, if $\sigma_k > 0$ for infinitely many indices $k \in K_2 \subset K_1$, the inequality (19) implies that, for $k \in K_2, k \geq k_0$,
\[
\langle g(x_k - \xi_k \alpha_k^+ \sigma_k F(x_k)), F(x_k) \rangle \leq \frac{C\gamma \alpha_k^+}{\sigma_k} \leq \frac{C\gamma \alpha_k^+}{\sigma_{\min}}.
\] (20)

Using (17) and proceeding in the same way, we obtain that, for $k \in K_2, k \geq k_0$,
\[
\langle g(x_k + \xi_k' \alpha_k^- \sigma_k F(x_k)), F(x_k) \rangle \geq -\frac{C\gamma \alpha_k^-}{\sigma_k} \geq -\frac{C\gamma \alpha_k^-}{\sigma_{\min}}
\] (21)

for some $\xi_k' \in [0, 1]$.

Since $\alpha_k^+ \rightarrow 0$, $\alpha_k^- \rightarrow 0$, and $\|\sigma_k F(x_k)\|$ is bounded, taking limits in (20) and (21), we obtain that
\[
\langle g(x_*), F(x_*) \rangle = 0.
\] (22)

If $\sigma_k < 0$ for infinitely many indices, proceeding in an analogous way, we also deduce (22). Thus, the thesis is proved. \hfill \Box

**Corollary 1.** Assume that $\{x_k\}_{k \in \mathbb{N}}$ is generated by the DF-SANE algorithm, $x_*$ is a limit point of $\{x_k\}_{k \in K}$ and for all $v \in \mathbb{R}^n, v \neq 0$,
\[
\langle J(x_*)v, v \rangle \neq 0.
\]

Then, $F(x_*) = 0$.

**Proof.** Straightforward, using Theorem 1. \hfill \Box

As usually, we say that a matrix $A \in \mathbb{R}^{m \times n}$ is positive-definite if $\langle Av, v \rangle > 0$ for all $v \in \mathbb{R}^n, v \neq 0$. If $J(x)$ is positive-definite for all $x \in \mathbb{R}^n$ we say that the mapping $F$ is strictly monotone. If $F$ is strictly monotone or $-F$ is strictly monotone, we say that the mapping $F$ is strict. If a mapping is strict and admits a solution, its solution must be unique. See [29], Chapter 5.

**Corollary 2.** Assume that $\{x_k\}_{k \in \mathbb{N}}$ is generated by the DF-SANE algorithm and the mapping $F$ is strict. Then, every bounded subsequence of $\{x_k\}_{k \in K}$ converges to the solution of (1).

**Proof.** Straightforward, using Corollary 1. \hfill \Box

**Corollary 3.** Assume that $\{x_k\}_{k \in \mathbb{N}}$ is generated by the DF-SANE algorithm, the mapping $F$ is strict and the level set $\{x \in \mathbb{R}^n \mid f(x) \leq f(x_0) + \eta\}$ is bounded. Then, $\{x_k\}_{k \in K}$ converges to the solution of (1).

**Proof.** Straightforward, using Corollary 2. \hfill \Box

So far, we proved that at every limit point of $\{x_k\}_{k \in K}$ the gradient of $\|F(x_*)\|^2$ is orthogonal to the residual $F(x_*)$. The case in which there exists a limit point of $\{x_k\}_{k \in \mathbb{N}}$
at which $F(x_*) = 0$ deserves further analysis. The theorem below shows that, when such a limit point exists, then all the limit points of the sequence generated by the algorithm are solutions of the nonlinear system.

**Theorem 2.** Assume that the sequence $\{x_k\}_{k \in \mathbb{N}}$ is generated by the DF-SANE Algorithm and that there exists a limit point $x_*$ of $\{x_k\}_{k \in \mathbb{N}}$ such that $F(x_*) = 0$. Then

$$\lim_{k \to \infty} F(x_k) = 0.$$ 

Consequently, $F(x)$ vanishes at every limit point of $\{x_k\}_{k \in \mathbb{N}}$.

**Proof.** Let $K_1$ be an infinite subset of $\mathbb{N}$ such that

$$\lim_{k \in K_1} x_k = x_*$$

and

$$F(x_*) = 0. \quad (23)$$

Then,

$$\lim_{k \in K_1} F(x_k) = 0.$$ 

Therefore, since $x_{k+1} = x_k \pm \alpha_k \sigma_k F(x_k)$ and $|\alpha_k \sigma_k| \leq \sigma_{\text{max}}$ for all $k \in \mathbb{N}$,

$$\lim_{k \in K_1} \|x_{k+1} - x_k\| = 0.$$ 

So,

$$\lim_{k \in K_1} x_{k+1} = x_*.$$ 

Proceeding by induction, we may prove that for all fixed $\ell \in \{1, 2, \ldots, 2M\},$

$$\lim_{k \in K_1} x_{k+\ell} = x_* \quad (24)$$

Now, by (9), for all $k \in K_1$, we can choose $\mu(k) \in \{0, 1, \ldots, 2M - 1\}$ such that

$$k + \mu(k) \in K_+.$$ 

Moreover, the same $\mu(k)$ must be repeated infinitely many times. So, there exists $\ell_0 \in \{0, 1, \ldots, 2M - 1\}$ such that $\mu(k) = \ell_0$ for infinitely many indices $k \in K_1$. Consider

$$K_2 = \{k + \mu(k) \mid k \in K_1 \text{ and } \mu(k) = \ell_0\}.$$ 

By (24) and (25) we have that $K_2 \subset K_+$ and

$$\lim_{k \in K_2} x_k = x_*.$$ 

Then, by (23),

$$\lim_{k \in K_2} F(x_k) = 0.$$
Since $K_2 \subset K_+$, there exists $J_1$ an infinite subset of $\mathbb{N}$ such that
\[
\lim_{j \in J_1} x_{\nu(j)} = x_\ast \tag{26}
\]
and
\[
\lim_{j \in J_1} f(x_{\nu(j)}) = \lim_{j \in J_1} V_j = 0. \tag{27}
\]

Let us write $J_1 = \{j_1, j_2, j_3, \ldots\}$, where $j_1 < j_2 < j_3 < \ldots$ and $\lim_{i \to \infty} j_i = \infty$. By (27) we have that
\[
\lim_{i \to \infty} V_{j_i} = 0. \tag{28}
\]

Now, by (5) we have that for all $j \in \mathbb{N}$, $j > j_i$,
\[
V_j \leq V_{j_i} + \sum_{\ell = M_{j_i}}^{\infty} \eta_{\ell}. \tag{29}
\]

Therefore,
\[
\sup_{j \geq j_i} V_j \leq V_{j_i} + \sum_{\ell = M_{j_i}}^{\infty} \eta_{\ell}. \tag{29}
\]

By the summability of $\eta_k$,
\[
\lim_{i \to \infty} \sum_{\ell = M_{j_i}}^{\infty} \eta_{\ell} = 0.
\]

Then, by (28), taking limits on both sides of (29), we get:
\[
\lim_{i \to \infty} \sup_{j \geq j_i} V_j = 0.
\]

Thus,
\[
\lim_{j \to \infty} V_j = 0.
\]

By the definition of $V_j$ this implies that
\[
\lim_{k \to \infty} \|F(x_k)\| = \lim_{k \to \infty} f(x_k) = 0, \tag{30}
\]
as we wanted to prove.

The second part of the proof is straightforward: if $\bar{x}$ is a limit point of $\{x_k\}_{k \in \mathbb{N}}$ there exists a subsequence $\{x_k\}_{k \in K_3}$ that converges to $\bar{x}$. By (30) and the continuity of $F$ we have that
\[
F(\bar{x}) = \lim_{k \in K_3} F(x_k) = 0.
\]

Now we prove two theorems of local convergence type. Theorem 3 says that if an isolated solution is a limit point of $\{x_k\}_{k \in \mathbb{N}}$, then the whole sequence $x_k$ converges to this solution.

**Theorem 3.** Assume that the sequence $\{x_k\}_{k \in \mathbb{N}}$ is generated by the DF-SANE algorithm and that there exists a limit point $x_\ast$ of the sequence $\{x_k\}_{k \in \mathbb{N}}$ such that $F(x_\ast) = 0$. Moreover, assume that there exists $\delta > 0$ such that $F(x) \neq 0$ whenever $0 < \|x - x_\ast\| \leq \delta$. Then,
\[ \lim_{k \to \infty} x_k = x^*. \]

**Proof.** By Theorem 2 we have that

\[ \lim_{k \to \infty} F(x_k) = 0. \]

Therefore, since \( \alpha_k \) and \( \sigma_k \) are bounded,

\[ \lim_{k \to \infty} \|x_{k+1} - x_k\| = 0. \]

Thus, there exists \( k_1 \in \mathbb{N} \) such that

\[ \|x_{k+1} - x_k\| \leq \frac{\delta}{2} \quad \text{for all } k \geq k_1. \quad (31) \]

Consider the set

\[ S = \{ x \in \mathbb{R}^n \mid \frac{\delta}{2} \leq \|x - x^*\| \leq \delta \}. \]

By hypothesis, \( S \) does not contain any solution of \( F(x) = 0 \). But, by Theorem 2, all the limit points of \( \{x_k\}_{k \in \mathbb{N}} \) are solutions of (1). Therefore, \( S \) does not contain any limit point of \( \{x_k\}_{k \in \mathbb{N}} \). Thus, since \( S \) is compact, it cannot contain infinitely many iterates \( x_k \). This implies that there exists \( k_2 \in \mathbb{N} \) such that

\[ x_k \notin S \quad \text{for all } k \geq k_2. \quad (32) \]

Let \( k_3 \geq \max\{k_1, k_2\} \) be such that

\[ \|x_{k_3} - x^*\| \leq \frac{\delta}{2}. \]

By (31), we have:

\[ \|x_{k_3+1} - x^*\| \leq \|x_{k_3} - x^*\| + \|x_{k_3+1} - x_{k_3}\| \leq \delta. \]

But, by (32), \( x_{k_3+1} \notin S \), therefore, we have that

\[ \|x_{k_3+1} - x^*\| \leq \frac{\delta}{2}. \]

Continuing this argument inductively we have that

\[ \|x_k - x^*\| \leq \frac{\delta}{2} \quad \text{for all } k \geq k_3. \quad (33) \]

This implies that all the limit points \( \bar{x} \) of the sequence \( \{x_k\}_{k \in \mathbb{N}} \) are such that

\[ \|\bar{x} - x^*\| \leq \frac{\delta}{2}. \]

By Theorem 2, \( F(\bar{x}) = 0 \) at every limit point \( \bar{x} \) and, by the hypothesis of this theorem, the set defined by \( 0 < \|x - x^*\| \leq \frac{\delta}{2} \) does not contain solutions of (1). Therefore, this set does not contain limit points. So, the only limit point \( \bar{x} \) that satisfies \( \|x - x^*\| \leq \frac{\delta}{2} \) is \( x^* \). So, by (33), the sequence converges to \( x^* \) as we wanted to prove. \( \square \)
Theorem 4 is our second local convergence theorem. For proving it we need a new definition and a technical lemma. We say that \( x^* \in \mathbb{R}^n \) is a strongly isolated solution of (1) if \( F(x^*) = 0 \) and there exists \( \varepsilon > 0 \) such that
\[
0 < \|x - x^*\| \leq \varepsilon \Rightarrow \langle J(x)F(x), F(x) \rangle \neq 0.
\]
That is, in a reduced neighborhood of a strongly isolated solution the residual \( F(x) \) is not orthogonal to the gradient \( J(x)^tF(x) \). Theorem 4 says that, if the initial point \( x_0 \) is close enough to an strongly isolated solution \( x^* \), then the sequence \( \{x_k\} \) converges to \( x^* \).

Observe that this cannot be deduced from Theorem 3 and, moreover, Theorem 3 cannot be deduced from this result either, since the strong isolation assumption is not necessary to prove that theorem.

Lemma 1. Assume that \( F(x^*) = 0 \), and \( k_0 \in \mathbb{N} \). Then, for all \( \delta_1 > 0 \), there exists \( \delta \in (0, \delta_1] \) such that for any possible initial point \( x_0 \) such that \( \|x_0 - x^*\| < \delta \), the \( k_0 \)-th iterate computed by the DF-SANE algorithm will satisfy
\[
\|x_{k_0} - x^*\| < \delta_1
\]
independently of the choices of \( \alpha_k, \sigma_k \) for \( k = 0, 1, \ldots, k_0 - 1 \).

Proof. We proceed by induction. If \( k_0 = 0 \) the result is trivial with \( \delta = \delta_1 \). Assume that it is true for \( k = 0, 1, \ldots, k_0 \) and let us prove it with for \( k_0 + 1 \). Let \( \delta_1 > 0 \). By the continuity of \( F \), since \( F(x^*) = 0 \) and \( |\sigma_k \alpha_k| \leq \sigma_{\text{max}} \), there exists
\[
\delta_2 \in (0, \delta_1/2)
\]
such that
\[
\|x_{k_0} - x^*\| < \delta_2 \Rightarrow \|x_{k_0+1} - x_{k_0}\| < \delta_1/2.
\]
But, by the inductive hypothesis, there exists \( \delta \in (0, \delta_2] \) such that
\[
\|x_0 - x^*\| < \delta \Rightarrow \|x_{k_0} - x^*\| < \delta_2.
\]
By (34), (35) and (36), if \( \|x_0 - x^*\| < \delta \), we have:
\[
\|x_{k_0+1} - x^*\| \leq \|x_{k_0} - x^*\| + \|x_{k_0+1} - x_{k_0}\| < \delta_2 + \delta_1/2 < \delta_1.
\]
This completes the proof.

Theorem 4. Assume that the sequence \( \{x_k\}_{k \in \mathbb{N}} \) is generated by the DF-SANE algorithm and \( x^* \) is a strongly isolated solution. Then, there exists \( \delta > 0 \) such that
\[
\|x_0 - x^*\| < \delta \Rightarrow \lim_{k \to \infty} x_k = x^*.
\]

Proof. Let \( \varepsilon > 0 \) be such that
\[
0 < \|x - x^*\| \leq \varepsilon \Rightarrow \langle J(x)F(x), F(x) \rangle \neq 0.
\]
Since $|\sigma_k \alpha_k| \leq \sigma_{\text{max}}$, $\|x_{k+1} - x_k\| \leq |\sigma_k \alpha_k| \|F(x_k)\|$ and $F(x_*) = 0$, the continuity of $F$ implies that there exists

$\varepsilon_1 \in (0, \varepsilon/2]$ such that

$$\|x_k - x_*\| \leq \varepsilon_1 \Rightarrow \|x_{k+1} - x_k\| < \varepsilon/2.$$  (38)

Define

$$C_\varepsilon = \{ x \in \mathbb{R}^n \mid \varepsilon_1 \leq \|x - x_*\| \leq \varepsilon \}.$$ 

Since $C_\varepsilon$ is compact and $f$ is continuous, there exists $\hat{x} \in C_\varepsilon$, $\beta > 0$, such that

$$\beta \equiv f(\hat{x}) \leq f(x) \text{ for all } x \in C_\varepsilon.$$  (39)

Since $f$ is continuous, the set $\{ x \in B(x_*, \varepsilon_1/2) \mid f(x) < \beta/2 \}$ is an open neighborhood of $x_*$. Therefore, there exists $\delta_1 \in (0, \varepsilon_1/2)$ such that

$$\|x - x_*\| \leq \delta_1 \Rightarrow f(x) < \beta/2.$$  (40)

Let $\bar{k}_0 \in K_+$ be such that

$$\sum_{k=0}^{\infty} \eta_k < \beta/2.$$  (41)

By Proposition 2, there exists $k_0 \geq \bar{k}_0$ ($k_0 \leq \bar{k}_0 + M$) such that

$$f(x_k) \leq f(x_{k_0}) + \frac{\beta}{2} \text{ for all } k \geq k_0.$$  (42)

Applying Lemma 1 for $\bar{k}_0$ and $k_0$, we deduce that there exists $\delta \in (0, \delta_1)$ such that

$$\|x_{k_0} - x_*\| \leq \delta \Rightarrow \|x_{k_0+j} - x_*\| < \varepsilon_1.$$  (43)

and

$$\|x_{k_0+j} - x_*\| \leq \delta \Rightarrow \|x_{k_0+j-1} - x_*\| < \varepsilon_1.$$  (44)

By (40) and (43), $f(x_{k_0}) < \beta/2$. So, by (42),

$$f(x_k) < \beta \text{ for all } k \geq k_0.$$  (45)

Let us prove by induction that, choosing $\|x_0 - x_*\| \leq \delta$,

$$\|x_{k_0+j} - x_*\| < \varepsilon_1$$  (46)

for all $j \in \mathbb{N}$. By (44), we have that (46) is true for $j = 0$.

Assume, as inductive hypothesis, that, for some $j \geq 1$,

$$\|x_{k_0+j-1} - x_*\| < \varepsilon_1.$$  (47)

Then,

$$\|x_{k_0+j-1} - x_*\| < \varepsilon/2.$$  (48)

But, by (38),

$$\|x_{k_0+j} - x_*\| \leq \|x_{k_0+j-1} - x_*\| + \|x_{k_0+j} - x_{k_0+j-1}\| < \varepsilon/2 + \varepsilon/2 = \varepsilon.$$  (49)
Since, by (45), $f(x_{k_0 + j}) < \beta$, (39) and (47) imply that $\|x_{k_0 + j} - x^*\| \leq \varepsilon_1$. This completes the inductive proof.

So, \{x_k\}_{k \geq k_0} \subset B(x_*, \varepsilon_1). Therefore, all the limit points $\bar{x}$ of \{x_k\}_{k \in \mathbb{N}} are such that $\|\bar{x} - x^*\| \leq \varepsilon_1 < \varepsilon$. But, by (37) and Theorem 1, the only possible limit point is $x_*$. Therefore, $\lim_{k \to \infty} x_k = x_*$ as we wanted to prove. \hfill \Box

**Corollary 4.** Assume that the sequence \{x_k\}_{k \in \mathbb{N}} is generated by the DF-SANE algorithm and $J(x_*)$ is either positive definite or negative definite. Then, there exists $\delta > 0$ such that

$$
\|x_0 - x^*\| < \delta \Rightarrow \lim_{k \to \infty} x_k = x_*.
$$

**Proof.** Using the continuity of $J$ we obtain that $x_*$ is strongly isolated. Then, the thesis follows from Theorem 4. \hfill \Box

### 4 Numerical results

We implemented DF-SANE with the following parameters: $nexp = 2$, $\sigma_{\text{min}} = 10^{-10}$, $\sigma_{\text{max}} = 10^{10}$, $\sigma_0 = 1$, $\tau_{\text{min}} = 0.1$, $\tau_{\text{max}} = 0.5$, $\gamma = 10^{-4}$, $M = 10$, $\eta_k = \|F(x_0)\|/(1 + k)^2$ for all $k \in \mathbb{N}$.

The spectral steplength was computed by the formula

$$
\sigma_k = \frac{\langle s_k, s_k \rangle}{\langle s_k, y_k \rangle},
$$

where $s_k = x_{k+1} - x_k$ and $y_k = F(x_{k+1}) - F(x_k)$. Observe that $y_k = [\int_0^1 J(x_k + ts_k) dt] s_k$, so $\sigma_k$ is the inverse of the Rayleigh quotient

$$
\frac{\langle [\int_0^1 J(x_k + ts_k) dt] s_k, s_k \rangle}{\langle s_k, s_k \rangle}.
$$

However, if $|\sigma_k| \notin [\sigma_{\text{min}}, \sigma_{\text{max}}]$, we replace the spectral coefficient by

$$
\sigma_k = \begin{cases} 
1 & \text{if } \|F(x_k)\| > 1, \\
\|F(x_k)\|^{-1} & \text{if } 10^{-5} \leq \|F(x_k)\| \leq 1, \\
10^5 & \text{if } \|F(x_k)\| < 10^{-5}.
\end{cases}
$$

Since we use big values for $\sigma_{\text{max}}$ and $1/\sigma_{\text{min}}$, this replacement rarely occurs. In the few cases in which the replacement is necessary the first trial point is $x_k - F(x_k)$ if $\|F(x_k)\| \geq 1$. If $10^{-5} \leq \|F(x_k)\| \leq 1$ the step $\sigma_k$ is such that the distance between $x_k$ and the first trial point is equal to 1. When $\|F(x_k)\| < 10^{-5}$ we prefer to allow the distance between $x_k$ and the trial point to be smaller, choosing for $\sigma_k$ the fixed value $10^{-5}$.

For choosing $\alpha_{+\text{new}}$ and $\alpha_{-\text{new}}$ at Step 2, we proceed as follows. Given $\alpha_+ > 0$, we take $\alpha_{+\text{new}} > 0$ as

$$
\alpha_{+\text{new}} = \begin{cases} 
\tau_{\text{min}} \alpha_+ & \text{if } \alpha_t < \tau_{\text{min}} \alpha_+, \\
\tau_{\text{max}} \alpha_+ & \text{if } \alpha_t > \tau_{\text{max}} \alpha_+, \\
\alpha_t & \text{otherwise},
\end{cases}
$$

15
where
\[
\alpha_t = \frac{\alpha^2 + f(x_k)}{f(x_k + \alpha + d) + (2\alpha - 1)f(x_k)}.
\]

We use similar formulae for choosing \(\alpha_{\text{new}}\) as a function of \(\alpha_{\text{old}}\), \(f(x_k)\) and \(f(x_k + \alpha + d)\).

This parabolic model is similar to the one described in [21, pp.142-143], in which the Jacobian matrix at \(x_k\) is replaced by the identity matrix (see also [12]).

We also implemented SANE [22] with the following parameters: \(\gamma = 10^{-4}\), \(\varepsilon = 10^{-8}\), \(\sigma_1 = 0.1\), \(\sigma_2 = 0.5\), \(\alpha_0 = 1\), \(M = 10\), and \(\delta =\)
\[
\begin{align*}
1 & \quad \text{if } \|F(x_k)\| > 1, \\
\|F(x_k)\|^{-1} & \quad \text{if } 10^{-5} \leq \|F(x_k)\| \leq 1, \\
10^5 & \quad \text{if } \|F(x_k)\| < 10^{-5}.
\end{align*}
\]

Both in SANE and DF-SANE we stop the process when
\[
\frac{\|F(x_k)\|}{\sqrt{n}} \leq e_a + e_r \frac{\|F(x_0)\|}{\sqrt{n}},
\]
where \(e_a = 10^{-5}\) and \(e_r = 10^{-4}\).

We ran SANE and DF-SANE using a set of large-scale test problems. All the test problems are fully described in Appendix A.

The numerical results are shown in Tables 1, 2, and 3. We report only one failure, and use the symbol (*), when running problem 18 with \(n = 100\). In that case, FD-SANE fails because it generates a sequence that converges to a point \(\bar{x}\) at which \(F(\bar{x})^T g(\bar{x}) = 0\), but \(F(\bar{x}) \neq 0\) and \(g(\bar{x}) \neq 0\).

The results from Tables 1 and 2 are summarized in Table 3. In Table 3 we compare the performance (number of problems for which each method is a winner with respect to number of iterations, function evaluations and computer time) between SANE and DF-SANE. In Tables 1 and 2 we report the problem number and the dimension of the problem (Function(\(n\))), the number of iterations (IT), the number of function evaluations (including the additional functional evaluations that SANE uses for approximating directional derivatives) (FE), the number of backtracks (BK), and the CPU time in seconds (T). In SANE it is necessary to evaluate the directional derivative \(\langle F(x_k), J(x_k)^T F(x_k) \rangle\) at each iteration. Since we assume that the Jacobian is not easily available we use the fact that
\[
\langle F(x_k), J(x_k)^T F(x_k) \rangle = \langle J(x_k) F(x_k), F(x_k) \rangle
\]
and the approximation
\[
J(x_k) F(x_k) \approx \frac{F(x_k + tF(x_k)) - F(x_k)}{t}
\]
where \(t > 0\) is a small parameter. Therefore, computing the approximate directional derivative involves an additional function evaluation (included in FE) at each iteration.

Our results indicate that the new fully free-derivative scheme DF-SANE is competitive with the SANE algorithm, which in turn is preferable quite frequently to the Newton-Krylov methods: Newton - GMRES, Newton - BiCGSTAB, and Newton - TFQMR (see [22] for comparisons). In particular, when comparing SANE with Newton - GMRES (which was the Krylov-like method with the best performance in [22]) the summary results shown in Table 4 were obtained.
<table>
<thead>
<tr>
<th>Function(n)</th>
<th>IT</th>
<th>FE</th>
<th>BK</th>
<th>T</th>
<th>IT</th>
<th>FE</th>
<th>BK</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>1(1000)</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td>.010</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>.000</td>
</tr>
<tr>
<td>1(10000)</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>.060</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>.050</td>
</tr>
<tr>
<td>2(500)</td>
<td>6</td>
<td>14</td>
<td>1</td>
<td>.010</td>
<td>11</td>
<td>11</td>
<td>0</td>
<td>.000</td>
</tr>
<tr>
<td>2(2000)</td>
<td>2</td>
<td>7</td>
<td>1</td>
<td>.010</td>
<td>11</td>
<td>11</td>
<td>0</td>
<td>.030</td>
</tr>
<tr>
<td>3(100)</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td>.010</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>.000</td>
</tr>
<tr>
<td>3(500)</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>.000</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>.010</td>
</tr>
<tr>
<td>4(99)</td>
<td>130</td>
<td>335</td>
<td>69</td>
<td>.060</td>
<td>99</td>
<td>289</td>
<td>66</td>
<td>.060</td>
</tr>
<tr>
<td>4(999)</td>
<td>130</td>
<td>335</td>
<td>69</td>
<td>.611</td>
<td>101</td>
<td>325</td>
<td>71</td>
<td>.611</td>
</tr>
<tr>
<td>5(9)</td>
<td>23</td>
<td>59</td>
<td>12</td>
<td>.000</td>
<td>42</td>
<td>68</td>
<td>12</td>
<td>.000</td>
</tr>
<tr>
<td>5(49)</td>
<td>552</td>
<td>1942</td>
<td>424</td>
<td>.070</td>
<td>732</td>
<td>2958</td>
<td>660</td>
<td>.130</td>
</tr>
<tr>
<td>6(100)</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>.000</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>.000</td>
</tr>
<tr>
<td>6(10000)</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>.040</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>.060</td>
</tr>
<tr>
<td>7(100)</td>
<td>23</td>
<td>49</td>
<td>2</td>
<td>.010</td>
<td>23</td>
<td>29</td>
<td>2</td>
<td>.000</td>
</tr>
<tr>
<td>7(10000)</td>
<td>23</td>
<td>49</td>
<td>2</td>
<td>.581</td>
<td>23</td>
<td>29</td>
<td>2</td>
<td>.511</td>
</tr>
<tr>
<td>8(1000)</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>.000</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>.000</td>
</tr>
<tr>
<td>8(10000)</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>.030</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>.030</td>
</tr>
<tr>
<td>9(100)</td>
<td>6</td>
<td>12</td>
<td>0</td>
<td>.040</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>.020</td>
</tr>
<tr>
<td>9(1000)</td>
<td>6</td>
<td>12</td>
<td>0</td>
<td>3.826</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>2.063</td>
</tr>
<tr>
<td>10(100)</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>.000</td>
<td>2</td>
<td>12</td>
<td>1</td>
<td>.000</td>
</tr>
<tr>
<td>10(500)</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>.010</td>
<td>2</td>
<td>12</td>
<td>1</td>
<td>.010</td>
</tr>
<tr>
<td>11(99)</td>
<td>11</td>
<td>34</td>
<td>4</td>
<td>.000</td>
<td>17</td>
<td>49</td>
<td>7</td>
<td>.000</td>
</tr>
<tr>
<td>11(399)</td>
<td>11</td>
<td>34</td>
<td>4</td>
<td>.020</td>
<td>17</td>
<td>49</td>
<td>7</td>
<td>.030</td>
</tr>
<tr>
<td>12(1000)</td>
<td>6</td>
<td>14</td>
<td>2</td>
<td>.040</td>
<td>30</td>
<td>62</td>
<td>12</td>
<td>.180</td>
</tr>
<tr>
<td>12(10000)</td>
<td>5</td>
<td>12</td>
<td>2</td>
<td>.421</td>
<td>23</td>
<td>59</td>
<td>11</td>
<td>2.073</td>
</tr>
<tr>
<td>13(100)</td>
<td>3</td>
<td>8</td>
<td>1</td>
<td>.000</td>
<td>3</td>
<td>7</td>
<td>1</td>
<td>.010</td>
</tr>
<tr>
<td>13(1000)</td>
<td>4</td>
<td>10</td>
<td>1</td>
<td>.020</td>
<td>4</td>
<td>8</td>
<td>1</td>
<td>.010</td>
</tr>
<tr>
<td>14(2500)</td>
<td>11</td>
<td>25</td>
<td>1</td>
<td>.210</td>
<td>11</td>
<td>17</td>
<td>1</td>
<td>.160</td>
</tr>
<tr>
<td>14(10000)</td>
<td>12</td>
<td>28</td>
<td>1</td>
<td>1.082</td>
<td>12</td>
<td>20</td>
<td>1</td>
<td>.871</td>
</tr>
<tr>
<td>15(5000)</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td>.060</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>.050</td>
</tr>
<tr>
<td>15(15000)</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td>.230</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>.180</td>
</tr>
<tr>
<td>16(500)</td>
<td>14</td>
<td>29</td>
<td>1</td>
<td>.000</td>
<td>14</td>
<td>16</td>
<td>1</td>
<td>.010</td>
</tr>
<tr>
<td>16(2000)</td>
<td>16</td>
<td>32</td>
<td>0</td>
<td>.010</td>
<td>16</td>
<td>16</td>
<td>0</td>
<td>.010</td>
</tr>
<tr>
<td>17(100)</td>
<td>9</td>
<td>19</td>
<td>1</td>
<td>.010</td>
<td>9</td>
<td>11</td>
<td>1</td>
<td>.000</td>
</tr>
<tr>
<td>17(1000)</td>
<td>7</td>
<td>15</td>
<td>1</td>
<td>.030</td>
<td>7</td>
<td>9</td>
<td>1</td>
<td>.030</td>
</tr>
<tr>
<td>18(50)</td>
<td>24</td>
<td>50</td>
<td>2</td>
<td>.010</td>
<td>19</td>
<td>21</td>
<td>1</td>
<td>.000</td>
</tr>
<tr>
<td>18(100)</td>
<td>24</td>
<td>49</td>
<td>1</td>
<td>.000</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>19(1000)</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td>.010</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>.010</td>
</tr>
<tr>
<td>19(50000)</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td>.771</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>.611</td>
</tr>
<tr>
<td>20(100)</td>
<td>32</td>
<td>67</td>
<td>2</td>
<td>.010</td>
<td>40</td>
<td>42</td>
<td>1</td>
<td>.010</td>
</tr>
<tr>
<td>20(1000)</td>
<td>51</td>
<td>117</td>
<td>9</td>
<td>.100</td>
<td>44</td>
<td>62</td>
<td>5</td>
<td>.070</td>
</tr>
<tr>
<td>21(399)</td>
<td>4</td>
<td>9</td>
<td>1</td>
<td>.010</td>
<td>5</td>
<td>7</td>
<td>1</td>
<td>.000</td>
</tr>
<tr>
<td>21(9999)</td>
<td>4</td>
<td>9</td>
<td>1</td>
<td>.200</td>
<td>5</td>
<td>7</td>
<td>1</td>
<td>.190</td>
</tr>
<tr>
<td>22(1000)</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>.000</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>.000</td>
</tr>
<tr>
<td>22(15000)</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>.030</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>.040</td>
</tr>
</tbody>
</table>

Table 1: SANE vs. DF-SANE for the first set of test problems.
<table>
<thead>
<tr>
<th>Function(n)</th>
<th>SANE</th>
<th>DF-SANE</th>
</tr>
</thead>
<tbody>
<tr>
<td>23(500)</td>
<td>1 10 1 .000</td>
<td>2 18 1 .010</td>
</tr>
<tr>
<td>23(1000)</td>
<td>1 11 1 .000</td>
<td>2 20 1 .000</td>
</tr>
<tr>
<td>24(500)</td>
<td>25 54 4 .030</td>
<td>54 199 18 .070</td>
</tr>
<tr>
<td>24(1000)</td>
<td>265 915 159 .951</td>
<td>17 25 3 .030</td>
</tr>
<tr>
<td>25(100)</td>
<td>2 6 1 .000</td>
<td>2 6 1 .000</td>
</tr>
<tr>
<td>25(500)</td>
<td>3 9 1 .000</td>
<td>3 9 1 .000</td>
</tr>
<tr>
<td>26(1000)</td>
<td>1 2 0 .000</td>
<td>1 1 0 .000</td>
</tr>
<tr>
<td>26(10000)</td>
<td>1 2 0 .020</td>
<td>1 1 0 .020</td>
</tr>
<tr>
<td>27(50)</td>
<td>10 20 0 .260</td>
<td>10 10 0 .140</td>
</tr>
<tr>
<td>27(100)</td>
<td>11 22 0 1.072</td>
<td>11 11 0 .561</td>
</tr>
<tr>
<td>28(100)</td>
<td>1 2 0 .000</td>
<td>1 1 0 .000</td>
</tr>
<tr>
<td>28(1000)</td>
<td>1 2 0 .000</td>
<td>1 1 0 .000</td>
</tr>
<tr>
<td>29(100)</td>
<td>1 4 1 .010</td>
<td>1 5 1 .000</td>
</tr>
<tr>
<td>29(1000)</td>
<td>1 4 1 .010</td>
<td>1 5 1 .010</td>
</tr>
<tr>
<td>30(99)</td>
<td>18 39 3 .000</td>
<td>11 16 2 .000</td>
</tr>
<tr>
<td>30(9999)</td>
<td>18 39 3 .791</td>
<td>11 16 2 .411</td>
</tr>
<tr>
<td>31(1000)</td>
<td>4 9 0 .030</td>
<td>6 6 0 .020</td>
</tr>
<tr>
<td>31(5000)</td>
<td>4 9 0 .160</td>
<td>6 6 0 .130</td>
</tr>
<tr>
<td>32(500)</td>
<td>6 12 0 .010</td>
<td>6 7 0 .010</td>
</tr>
<tr>
<td>32(1000)</td>
<td>6 12 0 .020</td>
<td>6 7 0 .020</td>
</tr>
<tr>
<td>33(1000)</td>
<td>3 20 2 .050</td>
<td>37 50 3 .120</td>
</tr>
<tr>
<td>33(5000)</td>
<td>3 22 1 .270</td>
<td>4 16 2 .230</td>
</tr>
<tr>
<td>34(1000)</td>
<td>22 52 4 .110</td>
<td>78 155 26 .381</td>
</tr>
<tr>
<td>34(5000)</td>
<td>12 27 1 .361</td>
<td>12 18 1 .280</td>
</tr>
<tr>
<td>35(1000)</td>
<td>21 45 2 .180</td>
<td>21 27 2 .110</td>
</tr>
<tr>
<td>35(5000)</td>
<td>29 63 3 1.402</td>
<td>38 48 3 1.202</td>
</tr>
<tr>
<td>36(1000)</td>
<td>21 45 2 .270</td>
<td>28 34 2 .210</td>
</tr>
<tr>
<td>36(5000)</td>
<td>44 96 7 2.954</td>
<td>26 36 4 1.272</td>
</tr>
<tr>
<td>37(1000)</td>
<td>23 49 2 .010</td>
<td>26 38 5 .010</td>
</tr>
<tr>
<td>37(5000)</td>
<td>23 49 2 .140</td>
<td>26 38 5 .210</td>
</tr>
<tr>
<td>38(1000)</td>
<td>19 40 2 .050</td>
<td>25 30 2 .040</td>
</tr>
<tr>
<td>38(5000)</td>
<td>19 40 2 .320</td>
<td>25 30 2 .340</td>
</tr>
<tr>
<td>39(1000)</td>
<td>55 126 13 .160</td>
<td>14 20 1 .030</td>
</tr>
<tr>
<td>39(5000)</td>
<td>55 126 13 1.041</td>
<td>14 20 1 .210</td>
</tr>
<tr>
<td>40(1000)</td>
<td>1 2 0 .000</td>
<td>1 1 0 .000</td>
</tr>
<tr>
<td>40(5000)</td>
<td>1 2 0 .020</td>
<td>1 1 0 .020</td>
</tr>
<tr>
<td>41(500)</td>
<td>7 15 1 .010</td>
<td>7 9 1 .010</td>
</tr>
<tr>
<td>41(1000)</td>
<td>2 5 1 .010</td>
<td>3 3 0 .000</td>
</tr>
<tr>
<td>42(1000)</td>
<td>110 268 45 .190</td>
<td>173 412 85 .330</td>
</tr>
<tr>
<td>42(5000)</td>
<td>110 268 45 1.392</td>
<td>173 412 85 2.654</td>
</tr>
<tr>
<td>43(100)</td>
<td>80 175 11 .010</td>
<td>86 108 9 .010</td>
</tr>
<tr>
<td>43(500)</td>
<td>488 1704 348 .601</td>
<td>586 1162 193 .451</td>
</tr>
<tr>
<td>44(1000)</td>
<td>2 4 0 .020</td>
<td>4 4 0 .030</td>
</tr>
<tr>
<td>44(5000)</td>
<td>2 4 0 .100</td>
<td>3 3 0 .090</td>
</tr>
</tbody>
</table>

Table 2: SANE vs. DF-SANE for the second set of test problems.


<table>
<thead>
<tr>
<th>Method</th>
<th>IT</th>
<th>FE</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>SANE</td>
<td>37</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>DF-SANE</td>
<td>10</td>
<td>64</td>
<td>38</td>
</tr>
<tr>
<td>Undecided</td>
<td>41</td>
<td>5</td>
<td>30</td>
</tr>
</tbody>
</table>

Table 3: Winners with respect to iterations, evaluations and time.

<table>
<thead>
<tr>
<th>Method</th>
<th>IT</th>
<th>FE</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newton-GMRES</td>
<td>51</td>
<td>9</td>
<td>19</td>
</tr>
<tr>
<td>SANE</td>
<td>9</td>
<td>51</td>
<td>41</td>
</tr>
</tbody>
</table>

Table 4: Winners with respect to iterations, evaluations and time between SANE and Newton-GMRES reported in [22].

5 Conclusions

The algorithm presented in this paper may be considered a damped quasi-Newton method for solving nonlinear systems. See [12, 26]. The iterations are

\[ x_{k+1} = x_k - \alpha_k B_k^{-1} F(x_k) \] (49)

where the Jacobian approximation \( B_k \) has the very simple form

\[ B_k = \frac{1}{\sigma_k} I. \] (50)

In most cases,

\[ \sigma_k = \frac{\langle s_k, s_k \rangle}{\langle y_k, s_k \rangle}. \] (51)

Due to the simplicity of the Jacobian approximation, the method is very easy to implement, memory requirements are minimal and, so, its use for solving large-scale nonlinear systems is attractive.

In [22] it was shown that, perhaps surprisingly, a procedure that obeys the scheme (49)–(51) behaves reasonably well for solving a number of classical nonlinear systems, most of them coming from discretization of boundary value problems. However, the algorithm introduced in [22] is not completely satisfactory in the sense that a directional derivative estimate is needed in order to ensure convergence and even well-definiteness of each iteration. In the present research we overcome that difficulty introducing the method DF - SANE, which does not need directional derivatives at all.

We were able to prove several convergence results:

1. There exists an infinite set of indices \( K \subset \mathbb{N} \) such that at every limit point of the subsequence \( \{ x_k \}_{k \in K} \), the gradient of \( \| F(x) \|^2 \) is orthogonal to the residual \( F(x) \). Therefore, if \( \| F(x) \| \) has bounded level sets, there exists a limit point \( x_* \) of \( \{ x_k \}_{k \in \mathbb{N}} \) such that \( \langle J(x_*)F(x_*), F(x_*) \rangle = 0 \).
2. If some limit point of \( \{x_k\}_{k \in \mathbb{N}} \) is a solution of (1), then every limit point is a solution;

3. If a limit point \( x_* \) of \( \{x_k\}_{k \in \mathbb{N}} \) is an isolated solution then the whole sequence converges to \( x_* \);

4. If the initial point \( x_0 \) is close enough to some strongly isolated solution \( x_* \) then the whole sequence converges to \( x_* \).

These results are obtained without using the specific formula of \( \sigma_k \) employed in our experiments. However, the method does not behave well for every choice of \( \sigma_k \). Therefore, much has to be said, from the theoretical point of view, to explain the behavior of algorithms associated to the safeguarded spectral choice of the steplength used here. In particular, although the theoretical properties of the Barzilai-Borwein method for minimizing convex quadratics are now well understood (see [30]), nothing is known about the properties of the spectral residual method for solving nonsymmetric linear systems. Global and local convergence theorems, as the ones presented in this paper, smooth the path for proving results on the order of convergence. Nevertheless, it is necessary to understand what happens in the linear case first.

Since the spectral residual method is a quasi-Newton method where the Jacobian approximation is a multiple of the identity matrix, the best behavior of this method must be expected when true Jacobians are close to matrices of that type. The analogy with the Barzilai-Borwein method allows one to conjecture in which (more general) situations the method should behave well. If the Jacobians are close to symmetric matrices with clustered eigenvalues (see [27]) a good behavior of the Barzilai-Borwein method can be predicted and, so, we also predict a fine behavior of the spectral residual method. Very likely, in many practical situations the performance of the method should be improved using some kind of preconditioning that transforms the Jacobian on a matrix with a small number of clusters of eigenvalues. So, with respect to preconditioning features, the situation is analogous to the one of Krylov-subspace methods. Preconditioned spectral gradient method for minimization were introduced in [25].

Our first set of experiments are discretization of boundary value problems. In general, the Jacobians are positive definite, so that the mappings \( F \) are generally monotone, or even strictly monotone. According to the corollaries of Theorem 1 this favors the behavior of DF-SANE, but also favors the behavior of almost every nonlinear-system solver. Since we are not using preconditioning at all, in general eigenvalues are not clustered. In some problems the Jacobians are well conditioned and in some other problems they are not. Moreover, in some problems the Jacobian is singular at the solution. In principle ill-conditioning affects adversely both spectral methods as Krylov subspace methods.

The second set of 22 problems does not show special characteristics from the point of view of positive definiteness or conditioning. Moreover, some of these problems have many solutions. In principle, we do not have strong reasons to predict a good behavior of DF-SANE, therefore the rather robust and efficient performance of the new algorithm for solving these problems is a pleasantly surprising fact that needs theoretical explanation.

Acknowledgment

We are indebted to Raúl Vignau, for his careful reading of the first draft of this paper. We are also indebted to an anonymous referee.
Appendix A: Test functions

We now list the test functions

\[ F_1(x) = (f_1(x), \ldots, f_n(x))^t \]

and the associated initial guess \( x_0 \).

1. **Exponential function 1**

\[
\begin{align*}
    f_1(x) &= e^{x_1 - 1}, \\
    f_i(x) &= i \left( e^{x_i - 1} - x_i \right), \quad \text{for } i = 2, 3, \ldots, n.
\end{align*}
\]

Initial guess: \( x_0 = \left(\frac{n}{n-1}, \frac{n}{n-1}, \ldots, \frac{n}{n-1}\right)^t \).

2. **Exponential function 2**

\[
\begin{align*}
    f_1(x) &= e^{x_1 - 1}, \\
    f_i(x) &= i \left( e^{x_i} + x_{i-1} - 1 \right), \quad \text{for } i = 2, 3, \ldots, n.
\end{align*}
\]

Initial guess: \( x_0 = \left(\frac{1}{n^2}, \frac{1}{n^2}, \ldots, \frac{1}{n^2}\right)^t \).

3. **Exponential function**

\[
\begin{align*}
    f_i(x) &= \frac{i}{10} \left( 1 - x_i^2 - e^{-x_i^2} \right), \quad \text{for } i = 2, 3, \ldots, n-1, \\
    f_n(x) &= \frac{n}{10} \left( 1 - e^{-x_n^2} \right).
\end{align*}
\]

Initial guess: \( x_0 = \left(\frac{1}{4n^2}, \frac{2}{4n^2}, \ldots, \frac{n}{4n^2}\right)^t \).

4. **Diagonal function premultiplied by a quasi-orthogonal matrix (n is a multiple of 3)** [17, pp. 89-90]

For \( i = 1, 2, \ldots, n/3, \)
\[
\begin{align*}
    f_{3i-2}(x) &= 0.6x_{3i-2} + 1.6x_{3i-2}^3 - 7.2x_{3i-1}^2 + 9.6x_{3i-1} - 4.8, \\
    f_{3i-1}(x) &= 0.48x_{3i-2} - 0.72x_{3i-1}^3 + 3.24x_{3i-1}^2 - 4.32x_{3i-1} - x_{3i} + 0.2x_{3i}^3 + 2.16, \\
    f_{3i}(x) &= 1.25x_{3i} - 0.25x_{3i}^3.
\end{align*}
\]

Initial guess: \( x_0 = \left(-1, \frac{1}{2}, -1, \ldots, -1, \frac{1}{2}, -1\right)^t \).

5. **Extended Rosenbrock function (n is even)** [17, p. 89]

For \( i = 1, 2, \ldots, n/2, \)
\[
\begin{align*}
    f_{2i-1}(x) &= 10 \left( x_{2i} - x_{2i-1}^2 \right), \\
    f_{2i}(x) &= 1 - x_{2i-1}.
\end{align*}
\]

Initial guess: \( x_0 = (5, 1, \ldots, 5, 1)^t \).
6. Chandrasekhar’s H-equation [21, p. 198]

\[ F_6(H)(\mu) = H(\mu) - \left(1 - \frac{c}{2} \int_0^1 \frac{\mu H(\nu)}{\mu + \nu} d\mu\right)^{-1} = 0. \]

The discretized version is:

\[ f_i(x) = x_i - \left(1 - \frac{c}{2n} \sum_{j=1}^{n} \frac{\mu_i x_j}{\mu_i + \mu_j}\right)^{-1}, \quad \text{for } i = 1, 2, \ldots, n, \]

with \( c \in [0, 1) \) and \( \mu_i = (i - 1/2)/n \), for \( 1 \leq i \leq n \). (In our experiments we take \( c = 0.9 \)).

Initial guess: \( x_0 = (1, 1, \ldots, 1)^t \).

7. Badly scaled augmented Powell’s function \((n \text{ is a multiple of } 3)\) [17, p. 89]

For \( i = 1, 2, \ldots, n/3 \),

\[ f_{3i-2}(x) = 10^4 x_{3i-1} x_{3i-2} - 1, \]
\[ f_{3i-1}(x) = \exp(-x_{3i-2}) + \exp(-x_{3i-1}) - 1.0001, \]
\[ f_{3i}(x) = \phi(x_{3i}), \]

with

\[ \phi(t) = \begin{cases} 
0.5t - 2 & t \leq 1 \\
(-592t^3 + 888t^2 + 4551t - 1924) / 1998, & -1 < t < 2 \\
0.5t + 2 & t \geq 2.
\end{cases} \]

Initial guess: \( x_0 = (10^{-3}, 18, 1, 10^{-3}, 18, 1, \ldots)^t \).

8. Trigonometric function

\[ f_i(x) = 2 \left(n + i(1 - \cos x_i) - \sin x_i - \sum_{j=1}^{n} \cos x_j\right) (2 \sin x_i - \cos x_i). \]

Initial guess: \( x_0 = \left(\frac{101}{100}, \ldots, \frac{101}{100}\right)^t \).

9. Singular function

\[ f_1(x) = \frac{1}{3} x_1^3 + \frac{1}{2} x_2^2 \]
\[ f_i(x) = -\frac{1}{2} x_i^2 + \frac{i}{3} x_i^3 + \frac{1}{2} x_{i+1}^2, \quad \text{for } i = 2, 3, \ldots, n - 1, \]
\[ f_n(x) = -\frac{1}{2} x_n^2 + \frac{n}{3} x_n^3. \]

Initial guess: \( x_0 = (1, 1, \ldots, 1)^t \).

10. Logarithmic function

\[ f_i(x) = \ln(x_i + 1) - \frac{x_i}{n}, \quad \text{for } i = 1, 2, \ldots, n. \]

Initial guess: \( x_0 = (1, 1, \ldots, 1)^t \).
11. Broyden Tridiagonal function [18, pp. 471-472]

\[
\begin{align*}
f_1(x) &= (3 - 0.5x_1)x_1 - 2x_2 + 1, \\
f_i(x) &= (3 - 0.5x_i)x_i - x_{i-1} - 2x_{i+1} + 1, \quad \text{for } i = 2, 3, \ldots, n - 1, \\
f_n(x) &= (3 - 0.5x_n)x_n - x_{n-1} + 1. 
\end{align*}
\]

Initial guess: \(x_0 = (-1, -1, \ldots, -1)^t\).

12. Trigexp function [18, p. 473]

\[
\begin{align*}
f_1(x) &= 3x_1^2 + 2x_2 - 5 + \sin(x_1 - x_2)\sin(x_1 + x_2), \\
f_i(x) &= -x_{i-1}e^{(x_{i-1}-x_i)} + x_i(4 + 3x_i^2) + 2x_{i+1} \\
&\quad + \sin(x_i - x_{i+1})\sin(x_i + x_{i+1}) - 8, \quad \text{for } i = 2, 3, \ldots, n - 1, \\
f_n(x) &= -x_{n-1}e^{(x_{n-1}-x_n)} + 4x_n - 3.
\end{align*}
\]

Initial guess: \(x_0 = (0, 0, \ldots, 0)^t\).

13. Variable band function 1 [18, p. 474]

\[
\begin{align*}
f_1(x) &= -2x_1^2 + 3x_1 - 2x_2 + 0.5x_{\alpha_1} + 1, \\
f_i(x) &= -2x_i^2 + 3x_i - x_{i-1} - 2x_{i+1} + 0.5x_{\alpha_i} + 1, \quad \text{for } i = 2, 3, \ldots, n - 1, \\
f_n(x) &= -2x_n^2 + 3x_n - x_{n-1} + 0.5x_{\alpha_n} + 1,
\end{align*}
\]

and \(\alpha_i\) is a random integer number in \([\alpha_{\text{imin}}, \alpha_{\text{imax}}]\), where \(\alpha_{\text{imin}} = \max[1, i - 2]\) and \(\alpha_{\text{imax}} = \min[n, i + 2]\), for all \(i\).

Initial guess: \(x_0 = (0, 0, \ldots, 0)^t\).

14. Variable band function 2 [18, p. 474]

\[
\begin{align*}
f_1(x) &= -2x_1^2 + 3x_1 - 2x_2 + 0.5x_{\alpha_1} + 1, \\
f_i(x) &= -2x_i^2 + 3x_i - x_{i-1} - 2x_{i+1} + 0.5x_{\alpha_i} + 1, \quad \text{for } i = 2, 3, \ldots, n - 1, \\
f_n(x) &= -2x_n^2 + 3x_n - x_{n-1} + 0.5x_{\alpha_n} + 1,
\end{align*}
\]

and \(\alpha_i\) is a random integer number in \([\alpha_{\text{imin}}, \alpha_{\text{imax}}]\), where \(\alpha_{\text{imin}} = \max[1, i - 10]\) and \(\alpha_{\text{imax}} = \min[n, i + 10]\), for all \(i\).

Initial guess: \(x_0 = (0, 0, \ldots, 0)^t\).

15. Function 15 [18, p. 475]

\[
\begin{align*}
f_1(x) &= -2x_1^2 + 3x_1 + 3x_{n-4} - x_{n-3} - x_{n-2} + 0.5x_{n-1} - x_n + 1, \\
f_i(x) &= -2x_i^2 + 3x_i - x_{i-1} - 2x_{i+1} + 3x_{n-4} - x_{n-3} - x_{n-2} + 0.5x_{n-1} \\
&\quad - x_n + 1, \quad \text{for } i = 2, 3, \ldots, n - 1, \\
f_n(x) &= -2x_n^2 + 3x_n - x_{n-1} + 3x_{n-4} - x_{n-3} - x_{n-2} + 0.5x_{n-1} - x_n + 1.
\end{align*}
\]

Initial guess: \(x_0 = (-1, -1, \ldots, -1)^t\).
16. **Strictly convex function 1 [31, p. 29]**

   $F(x)$ is the gradient of $h(x) = \sum_{i=1}^{n} (e^{x_i} - x_i)$.

   $f_i(x) = e^{x_i} - 1$, for $i = 1, 2, \ldots, n$.

   Initial guess: $x_0 = \left( \frac{1}{n}, \frac{2}{n}, \ldots, 1 \right)^t$.

17. **Strictly convex function 2 [31, p. 30]**

   $F(x)$ is the gradient of $h(x) = \sum_{i=1}^{n} \frac{i}{10} (e^{x_i} - x_i)$.

   $f_i(x) = \frac{i}{10} (e^{x_i} - 1)$, for $i = 1, 2, \ldots, n$.

   Initial guess: $x_0 = (1, 1, \ldots, 1)^t$.

18. **Function 18 (n is a multiple of 3)**

   For $i = 1, 2, \ldots, n/3$,

   $f_{3i-2}(x) = x_{3i-2} x_{3i-1} - x_{3i}^2 - 1$,

   $f_{3i-1}(x) = x_{3i-2} x_{3i-1} x_{3i} - x_{3i}^2 - 2 + x_{3i-1} - 2$,

   $f_{3i}(x) = e^{-x_{3i-2}} - e^{-x_{3i-1}}$.

   Initial guess: $x_0 = (0, 0, \ldots, 0)^t$.

19. **Zero Jacobian function**

   $f_1(x) = \sum_{j=1}^{n} x_j^2$,

   $f_i(x) = -2x_1x_i$, for $i = 2, \ldots, n$.

   Initial guess: $x_0^1 = \frac{100(n-100)}{n}$, and for all $i \geq 2$, $x_0^i = \frac{(n-1000)(n-500)}{(60n)^2}$.

20. **Geometric programming function**

   $f_i(x) = \sum_{t=1}^{5} \left[ 0.2 t x_i^{0.2t-1} \prod_{k=1, k\neq i}^{n} x_k^{0.2t} \right]$.

   Initial guess: $x_0 = (1, 1, \ldots, 1)^t$.

21. **Function 21 (n is a multiple of 3)**

   For $i = 1, 2, \ldots, n/3$:

   $f_{3i-2}(x) = x_{3i-2} x_{3i-1} - x_{3i}^2 - 1$,

   $f_{3i-1}(x) = x_{3i-2} x_{3i-1} x_{3i} - x_{3i}^2 - 2 + x_{3i-1} - 2$,

   $f_{3i}(x) = e^{-x_{3i-2}} - e^{-x_{3i-1}}$.

   Initial guess: $x_0 = (1, 1, \ldots, 1)^t$. 

   24
22. **Linear function—full rank.**

\[
f_i(x) = x_i - (2/n) \sum_{j=1}^{n} x_j + 1.
\]

Initial guess: \( x_0 = (100, 100, \ldots, 100)^t \).

23. **Linear function—rank 2.**

\[
f_1(x) = x_1 - 1,
\]

\[
f_i(x) = i \sum_{j=1}^{n} jx_j - i, \text{ for } i = 2, 3, \ldots, n.
\]

Initial guess: \( x_0 = \left(1, \frac{1}{n}, \ldots, \frac{1}{n}\right)^t \).

24. **Penalty I function.**

\[
f_i(x) = \sqrt{10^{10} (x_i - 1)}, \text{ for } i = 1, 2, \ldots, n - 1,
\]

\[
f_n(x) = \left(\frac{1}{4n}\right) \sum_{j=1}^{n} x_j^2 - \frac{1}{4}.
\]

Initial guess: \( x_0 = \left(\frac{1}{3}, \frac{1}{3}, \ldots, \frac{1}{3}\right)^t \).

25. **Brown almost function.**

\[
f_i(x) = x_i + \sum_{j=1}^{n} x_j - (n + 1), \text{ for } i = 1, 2, \ldots, n - 1,
\]

\[
f_n(x) = \prod_{j=1}^{n} x_j - 1.
\]

Initial guess: \( x_0 = \left(\frac{n-1}{n}, \frac{n-1}{n}, \ldots, \frac{n-1}{n}\right)^t \).

26. **Variable dimensioned function.**

\[
f_i(x) = x_i - 1, \text{ for } i = 1, 2, \ldots, n - 2,
\]

\[
f_{n-1}(x) = \sum_{j=1}^{n-2} j(x_j - 1),
\]

\[
f_n(x) = \left(\sum_{j=1}^{n-2} j(x_j - 1)\right)^2.
\]

Initial guess: \( x_0 = \left(1 - \frac{1}{n}, 1 - \frac{2}{n}, \ldots, 0\right)^t \).
27. Geometric function.

\[ f_i(x) = \sum_{t=1}^{5} \left( \frac{t}{5} \right)^{x_{i}^{(t/5-1)}} \prod_{k=1, k \neq i}^{n} x_{k}^{t/5} \].

Initial guess: \( x_0 = (0.9, 0.9, \ldots, 0.9)^t \).

28. Extended Powel singular function \((n)\) is a multiple of 4\) [28].

For \( i = 1, 2, \ldots, n/4, \)

\[ f_{4i-3}(x) = x_{4i-3} + 10x_{4i-2}, \]
\[ f_{4i-2}(x) = \sqrt{5}(x_{4i-1} - x_{4i}), \]
\[ f_{4i-1}(x) = (x_{4i-2} - 2x_{4i-1})^2, \]
\[ f_{4i}(x) = \sqrt{10}(x_{4i-3} - x_{4i})^2. \]

Initial guess: \( x_0 = (1.5 \times 10^{-4}, 1.5 \times 10^{-4}, \ldots, 1.5 \times 10^{-4})^t \).

29. Function 27.

\[ f_1(x) = \sum_{j=1}^{n} x_j^2, \]
\[ f_i(x) = -2x_1x_i, \text{ for } i = 2, 3, \ldots, n. \]

Initial guess: \( x_0 = \left(100, \frac{1}{n}, \ldots, \frac{1}{n} \right)^t. \)

30. Tridimensional valley function \((n)\) is a multiple of 3\) [16].

For \( i = 1, 2, \ldots, n/3, \)

\[ f_{3i-2}(x) = (c_2x_{3i-2}^3 + c_1x_{3i-2}) \exp \left( -\frac{x_{3i-2}^2}{100} \right) - 1, \]
\[ f_{3i-1}(x) = 10(\sin(x_{3i-2}) - x_{3i-1}), \]
\[ f_{3i}(x) = 10(\cos(x_{3i-2}) - x_{3i}), \]

donde
\[ c_1 = 1.003344481605351 \]
\[ c_2 = -3.344481605351171 \times 10^{-3} \]

Initial guess: \( x_0 = (2, 1, 2, 1, 2, \ldots)^t \).

31. Complementary function \((n)\) is even.

For \( i = 1, 2, \ldots, n/2, \)

\[ f_{2i-1}(x) = \left( x_{2i-1}^2 + \left( x_{2i-1}e^{x_{2i-1}} - \frac{1}{n} \right)^2 \right)^{1/2} - x_{2i-1} \]
\[ -x_{2i-1}e^{x_{2i-1}} + \frac{1}{n}, \]
\[ f_{2i}(x) = \left( x_{2i}^2 + (3x_i + \sin(x_{2i}) + e^{x_{2i}})^2 \right)^{1/2} - x_{2i} \]
\[ -3x_{2i} - \sin(x_{2i}) - e^{x_{2i}}. \]
32. **Minimal function.**

\[
f_i(x) = \frac{(\ln x_i + \exp(x_i)) - \sqrt{(\ln x_i - \exp(x_i))^2 + 10^{-10}}}{2}
\]

Initial guess: \(x_0 = (0.5, \ldots, 0.5)^t\).

33. **Hanbook function.**

\[
f_i(x) = 0.05(x_i - 1) + 2 \sin\left(\sum_{j=1}^{n} (x_j - 1) + \sum_{j=1}^{n} (x_j - 1)^2\right)
\]

\[
(1 + 2(x_i - 1)) + 2 \sin\left(\sum_{j=1}^{n} (x_j - 1)\right).
\]

Initial guess: \(x_0 = (1, \ldots, 1)^t\).

34. **Tridiagonal system** [23].

\[
f_1(x) = 4(x_1 - x_2^2),
\]

\[
f_i(x) = 8x_i(x_i^2 - x_{i-1}) - 2(1 - x_i) + 4(x_i - x_{i+1}^2), \text{ for } i = 2, \ldots, n - 1
\]

\[
f_n(x) = 8x_n(x_n^2 - x_{n-1}) - 2(1 - x_n).
\]

Initial guess: \(x_0 = (12, \ldots, 12)^t\).

35. **Five-diagonal system** [23].

\[
f_1(x) = 4(x_1 - x_2^2) + x_2 - x_3^2,
\]

\[
f_2(x) = 8x_2(x_2^2 - x_1) - 2(1 - x_2) + 4(x_2 - x_3^2) + x_3 - x_4^2,
\]

\[
f_i(x) = 8x_i(x_i^2 - x_{i-1}) - 2(1 - x_i) + 4(x_i - x_{i+1}^2) + x_{i-1}^2 - x_{i-2}^2
\]

\[
+ x_{i+1} - x_{i+2}^2, \text{ for } i = 3, \ldots, n - 2,
\]

\[
f_{n-1}(x) = 8x_{n-1}(x_{n-1}^2 - x_{n-2}) - 2(1 - x_{n-1}) + 4(x_{n-1} - x_n^2)
\]

\[
+ x_{n-2}^2 - x_{n-3},
\]

\[
f_n(x) = 8x_n(x_n^2 - x_{n-1}) - 2(1 - x_n) + x_{n-1}^2 - x_{n-2}.
\]

Initial guess: \(x_0 = (-2, \ldots, -2)^t\).

36. **Seven-diagonal system** [23].

\[
f_1(x) = 4(x_1 - x_2^2) + x_2 - x_3^2 + x_3 - x_4^2,
\]

\[
f_2(x) = 8x_2(x_2^2 - x_1) - 2(1 - x_2) + 4(x_2 - x_3^2) + x_3 - x_4^2
\]

\[
+ x_4 - x_5^2,
\]

\[
f_3(x) = 8x_3(x_3^2 - x_2) - 2(1 - x_3) + 4(x_3 - x_4^2) + x_4^2 - x_1 + x_4
\]

\[
- x_2^2 + x_1^2 + x_5 - x_6^2,
\]

\[
f_i(x) = 8x_i(x_i^2 - x_{i-1}) - 2(1 - x_i) + 4(x_i - x_{i+1}^2) + x_{i-1}^2
\]

\[
+ x_{i+1} - x_{i+2}^2, \text{ for } i = 3, \ldots, n - 2,
\]

\[
f_{n-1}(x) = 8x_{n-1}(x_{n-1}^2 - x_{n-2}) - 2(1 - x_{n-1}) + 4(x_{n-1} - x_n^2)
\]

\[
+ x_{n-2}^2 - x_{n-3},
\]

\[
f_n(x) = 8x_n(x_n^2 - x_{n-1}) - 2(1 - x_n) + x_{n-1}^2 - x_{n-2}.
\]
- \( x_{i-2} + x_{i+1} - x_{i+2}^2 + x_{i-2}^2 + x_{i+2} - x_{i-3} \)
- \( -x_{i+3}^2 \), for \( i = 4, \ldots, n - 3 \),

\( f_{n-2}(x) = 8x_{n-2}(x_{n-2}^2 - x_{n-3}) - 2(1 - x_{n-2}) + 4(x_{n-2} - x_{n-1}^2) \)
+ \( x_{n-3}^2 - x_{n-4} + x_{n-1} - x_n + x_{n-4} + x_n - x_{n-5} \),

\( f_{n-1}(x) = 8x_{n-1}(x_{n-1}^2 - x_{n-2}) - 2(1 - x_{n-1}) + 4(x_{n-1} - x_{n-2}^2) + x_{n-2}^2 - x_{n-3}^2 + x_n + x_{n-3} - x_{n-4} \),

\( f_n(x) = 8x_n(x_n^2 - x_{n-1}) - 2(1 - x_n) + x_{n-1}^2 - x_{n-2}^2 + x_{n-2}^2 - x_{n-3} \).

Initial guess: \( x_0 = (-3, \ldots, -3)^t \).

37. **Extended Freudenstein and Roth function** (\( n \) is even) [4].

For \( i = 1, 2, \ldots, n/2 \),

\( f_{2i-1}(x) = x_{2i-1} + ((5 - x_{2i})x_{2i} - 2)x_{2i} - 13 \),

\( f_{2i}(x) = x_{2i-1} + ((x_{2i} + 1)x_{2i} - 14)x_{2i} - 29 \).

Initial guess: \( x_0 = (6, 3, 6, 3, \ldots, 6, 3)^t \).

38. **Extended Cragg and Levy problem** (\( n \) is a multiple of 4) [28].

For \( i = 1, 2, \ldots, n/4 \),

\( f_{4i-3}(x) = (\exp(x_{4i-3}) - x_{4i-2})^2 \),

\( f_{4i-2}(x) = 10(x_{4i-2} - x_{4i-1})^3 \),

\( f_{4i-1}(x) = \tan^2(x_{4i-1} - x_{4i}) \),

\( f_{4i}(x) = x_{4i} - 1 \).

Initial guess: \( x_0 = (4, 2, 2, 2, 4, 2, 2, 2, \ldots)^t \).

39. **Extended Wood problem** (\( n \) is a multiple of 4) [20].

For \( i = 1, 2, \ldots, n/4 \),

\( f_{4i-3}(x) = -200x_{4i-3}(x_{4i-2}^2 - x_{4i-3}^2) - (1 - x_{4i-3}) \),

\( f_{4i-2}(x) = 200(x_{4i-2}^2 - x_{4i-3}^2) + 20(x_{4i-2} - 1) + 19.8(x_{4i} - 1) \),

\( f_{4i-1}(x) = -180x_{4i-1}(x_{4i} - x_{4i-1}^2) - (1 - x_{4i-1}) \),

\( f_{4i}(x) = 180(x_{4i} - x_{4i-1}^2) + 20.2(x_{4i} - 1) + 19.8(x_{4i-2} - 1) \).

Initial guess: \( x_0 = (0, \ldots, 0)^t \).

40. **Tridiagonal exponential problem** [4].

\( f_1(x) = x_1 - \exp(\cos(h(x_1 + x_2))) \),

\( f_i(x) = x_i - \exp(\cos(h(x_{i-1} + x_i + x_{i+1}))) \), for \( i = 2, \ldots, n - 1 \)

\( f_n(x) = x_n - \exp(\cos(h(x_{n-1} + x_n))) \),

\( h = 1/(n + 1) \).

Initial guess: \( x_0 = (1.5, \ldots, 1.5)^t \).
41. **Discrete boundary value problem** [28].

\[
\begin{align*}
   f_1(x) &= 2x_1 + 0.5h^2(x_1 + h)^3 - x_2, \\
   f_i(x) &= 2x_i + 0.5h^2(x_i + hi)^3 - x_{i-1} + x_{i+1}, \text{ for } i = 2, \ldots, n-1 \\
   f_n(x) &= 2x_n + 0.5h^2(x_n + hn)^3 - x_{n-1}, \\
   h &= 1/(n+1).
\end{align*}
\]

Initial guess: \( x_0 = (h(h-1), h(2h-1), \ldots, h(nh-1))^t \).

42. **Brent problem** [1].

\[
\begin{align*}
   f_1(x) &= 3x_1(x_2 - 2x_1) + x_2^2/4, \\
   f_i(x) &= 3x_i(x_{i+1} - 2x_i + x_{i-1}) + (x_{i+1} - x_{i-1})^2/4, \\
   &\quad \text{for } i = 2, \ldots, n-1 \\
   f_n(x) &= 3x_n(20 - 2x_n + x_{n-1}) + (20 - x_{n-1})^2/4, \\
   h &= 1/(n+1).
\end{align*}
\]

Initial guess: \( x_0 = (0, \ldots, 0, 20, 20)^t \).

43. **Troesch problem** [32].

\[
\begin{align*}
   f_1(x) &= 2x_1 + \rho h^2 \sinh(\rho x_1) - x_2, \\
   f_i(x) &= 2x_i + \rho h^2 \sinh(\rho x_i) - x_{i-1} - x_{i+1}, \text{ for } i = 2, \ldots, n-1 \\
   f_n(x) &= 2x_n + \rho h^2 \sinh(\rho x_n) - x_{n-1}, \\
   \rho &= 10, \quad h = 1/(n+1).
\end{align*}
\]

Initial guess: \( x_0 = (0, \ldots, 0)^t \).

44. **Trigonometric system** [33].

\[
\begin{align*}
   f_i(x) &= 5 - (l+1)(1 - \cos x_i) - \sin x_i - \sum_{j=5l+1}^{5l+5} \cos x_j, \\
   l &= \text{div}(i-1, 5).
\end{align*}
\]

Initial guess: \( x_0 = \left( \frac{1}{n}, \ldots, \frac{1}{n} \right)^t \).
Appendix B: Comparison of SANE algorithm with Newton-Krylov methods

We compare the performance of the SANE algorithm, on a set of large-scale test problems, with some recent implementations of inexact Newton methods. In particular we compare with Inexact Newton with GMRES (ING), Inexact Newton with Bi-CGSTAB (INBC), and Inexact Newton with TFQMR (INT). For all these techniques, an Armijo line search condition is used as a globalization strategy. For ING and INBC the Eisenstat-Walker formula is included [13]. The MATLAB code for all these methods was obtained from Kelley [21]. For all our experiments we use MATLAB 6.1 on a Pentium III personal computer at 700 MHz.

In the Appendix A we list the test functions, $F(x)$, and the associated initial guess, $x_0$, used in our experiments. The parameters used for inexact Newton methods are fully described in [21]. For the SANE algorithm we use $\gamma = 10^{-4}$, $\varepsilon = 10^{-8}$, $\sigma_1 = 0.1$, $\sigma_2 = 0.5$, $\alpha_0 = 1$, $M = 10$, $\delta = \begin{cases} 1 & \text{if } \|F_k\| > 1, \\ \|F_k\| & \text{if } 10^{-5} \leq \|F_k\| \leq 1, \\ 10^{-5} & \text{if } \|F_k\| < 10^{-5}, \end{cases}$ and the scalar $F_k^t J_k F_k$ is computed by

$$F_k^t J_k F_k \approx F_k^t \left[ \frac{F(x_k + hF_k) - F_k}{h} \right],$$

with $h = 10^{-7}$.

For choosing $\lambda$ at Step 7, we use the following parabolic model (Kelley [21, pp. 142-143]). Denoted by $\lambda_c > 0$ the current value of $\lambda$, we update $\lambda$ as follows

$$\lambda = \begin{cases} \sigma_1 \lambda_c & \text{if } \lambda_t < \sigma_1 \lambda_c, \\ \sigma_2 \lambda_c & \text{if } \lambda_t > \sigma_2 \lambda_c, \\ \lambda_t & \text{otherwise}, \end{cases}$$

where

$$\lambda_t = \frac{-\lambda_c^2 F(x_k)^t J(x_k) F(x_k)}{\|F(x_k + \lambda_c d_k)\|^2 - \|F(x_k)\|^2 - 2F(x_k)^t J(x_k) F(x_k) \lambda_c}.$$

In all our experiments we stop the process using (48), where $e_a = 10^{-6}$ and $e_r = 10^{-6}$. We claim that the method fails, and use the symbol (*), when some of the following options hold:

(a) the number of iterations is greater than or equal to 500; or

(b) the number of backtrackings at some line search is greater than or equal to 100.

The SANE algorithm could also fail if, at some iteration, a bad breakdown occurs, i.e., $|F_k^t J_k F_k| < \varepsilon \|F_k\|^2$, which will be reported with the symbol (**).

The numerical results are shown in Tables 5, 6, 7, and 8. We report the problem number and the dimension of the problem (Function(n)), the number of iterations (IT), the number of function evaluations (F), the number of backtrackings (BK), and the CPU time in seconds (T).
The results from Tables 5, 6, 7, and 8 are summarized in Table 9. In Table 9 we report the number of problems for which each method is a winner in number of iterations, number of function evaluations, and CPU time. We also report the failure percentage (FP) for each method.

In Table 10 we compare the performance (number of problems for which each method is a winner) between the SANE algorithm and Newton-GMRES, the best of the four competitors in function evaluations.

---

<table>
<thead>
<tr>
<th>Function(n)</th>
<th>IT</th>
<th>F</th>
<th>BK</th>
<th>T</th>
<th>Function(n)</th>
<th>IT</th>
<th>F</th>
<th>BK</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>1(1000)</td>
<td>5</td>
<td>42</td>
<td>0</td>
<td>0.09</td>
<td>11(500)</td>
<td>6</td>
<td>73</td>
<td>0</td>
<td>0.21</td>
</tr>
<tr>
<td>1(5000)</td>
<td>4</td>
<td>26</td>
<td>0</td>
<td>0.23</td>
<td>12(1000)</td>
<td>6</td>
<td>70</td>
<td>0</td>
<td>0.23</td>
</tr>
<tr>
<td>1(10000)</td>
<td>3</td>
<td>16</td>
<td>0</td>
<td>0.38</td>
<td>11(2000)</td>
<td>6</td>
<td>73</td>
<td>0</td>
<td>0.4</td>
</tr>
<tr>
<td>2(500)</td>
<td>5</td>
<td>51</td>
<td>0</td>
<td>0.14</td>
<td>12(100)</td>
<td>6</td>
<td>66</td>
<td>1</td>
<td>0.05</td>
</tr>
<tr>
<td>2(1000)</td>
<td>4</td>
<td>30</td>
<td>0</td>
<td>0.1</td>
<td>12(500)</td>
<td>6</td>
<td>66</td>
<td>1</td>
<td>0.21</td>
</tr>
<tr>
<td>2(2000)</td>
<td>4</td>
<td>29</td>
<td>0</td>
<td>0.16</td>
<td>12(1000)</td>
<td>6</td>
<td>64</td>
<td>1</td>
<td>0.23</td>
</tr>
<tr>
<td>3(50)</td>
<td>5</td>
<td>144</td>
<td>0</td>
<td>0.25</td>
<td>13(100)</td>
<td>6</td>
<td>67</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>3(100)</td>
<td>4</td>
<td>81</td>
<td>0</td>
<td>0.21</td>
<td>13(500)</td>
<td>6</td>
<td>73</td>
<td>2</td>
<td>0.33</td>
</tr>
<tr>
<td>3(200)</td>
<td>3</td>
<td>28</td>
<td>0</td>
<td>0.05</td>
<td>13(1000)</td>
<td>6</td>
<td>73</td>
<td>2</td>
<td>0.5</td>
</tr>
<tr>
<td>4(99)</td>
<td>6</td>
<td>68</td>
<td>2</td>
<td>0.05</td>
<td>14(100)</td>
<td>6</td>
<td>68</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>4(399)</td>
<td>6</td>
<td>68</td>
<td>2</td>
<td>0.085</td>
<td>14(500)</td>
<td>6</td>
<td>74</td>
<td>2</td>
<td>0.35</td>
</tr>
<tr>
<td>4(999)</td>
<td>7</td>
<td>87</td>
<td>2</td>
<td>0.27</td>
<td>14(1000)</td>
<td>6</td>
<td>73</td>
<td>2</td>
<td>0.49</td>
</tr>
<tr>
<td>5(1000)</td>
<td>6</td>
<td>51</td>
<td>0</td>
<td>0.1</td>
<td>15(500)</td>
<td>6</td>
<td>61</td>
<td>0</td>
<td>0.12</td>
</tr>
<tr>
<td>5(5000)</td>
<td>6</td>
<td>51</td>
<td>0</td>
<td>0.36</td>
<td>15(1000)</td>
<td>6</td>
<td>61</td>
<td>0</td>
<td>0.17</td>
</tr>
<tr>
<td>5(10000)</td>
<td>6</td>
<td>51</td>
<td>0</td>
<td>0.7</td>
<td>15(5000)</td>
<td>6</td>
<td>61</td>
<td>0</td>
<td>0.6</td>
</tr>
<tr>
<td>6(100)</td>
<td>4</td>
<td>27</td>
<td>0</td>
<td>0.1</td>
<td>16(1000)</td>
<td>4</td>
<td>24</td>
<td>0</td>
<td>0.05</td>
</tr>
<tr>
<td>6(500)</td>
<td>4</td>
<td>27</td>
<td>0</td>
<td>3.05</td>
<td>16(10000)</td>
<td>4</td>
<td>24</td>
<td>0</td>
<td>0.38</td>
</tr>
<tr>
<td>6(1000)</td>
<td>4</td>
<td>27</td>
<td>0</td>
<td>6</td>
<td>16(50000)</td>
<td>4</td>
<td>24</td>
<td>0</td>
<td>1.9</td>
</tr>
<tr>
<td>7(9)</td>
<td>5</td>
<td>65</td>
<td>11</td>
<td>0.03</td>
<td>17(100)</td>
<td>8</td>
<td>225</td>
<td>0</td>
<td>0.19</td>
</tr>
<tr>
<td>7(99)</td>
<td>5</td>
<td>65</td>
<td>11</td>
<td>0.1</td>
<td>17(500)</td>
<td>8</td>
<td>316</td>
<td>0</td>
<td>2.2</td>
</tr>
<tr>
<td>7(399)</td>
<td>5</td>
<td>65</td>
<td>11</td>
<td>0.31</td>
<td>17(1000)</td>
<td>8</td>
<td>338</td>
<td>0</td>
<td>2.7</td>
</tr>
<tr>
<td>8(1000)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>18(399)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>8(5000)</td>
<td>8</td>
<td>330</td>
<td>4</td>
<td>5</td>
<td>18(999)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>8(10000)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>18(9999)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>9(2500)</td>
<td>13</td>
<td>249</td>
<td>0</td>
<td>0.75</td>
<td>19(100)</td>
<td>5</td>
<td>50</td>
<td>0</td>
<td>0.05</td>
</tr>
<tr>
<td>9(5000)</td>
<td>13</td>
<td>244</td>
<td>0</td>
<td>1.3</td>
<td>19(500)</td>
<td>10</td>
<td>120</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>9(10000)</td>
<td>12</td>
<td>204</td>
<td>0</td>
<td>2.25</td>
<td>19(1000)</td>
<td>10</td>
<td>120</td>
<td>0</td>
<td>0.17</td>
</tr>
<tr>
<td>10(5000)</td>
<td>5</td>
<td>35</td>
<td>0</td>
<td>0.27</td>
<td>20(50)</td>
<td>14</td>
<td>224</td>
<td>0</td>
<td>2.343</td>
</tr>
<tr>
<td>10(10000)</td>
<td>5</td>
<td>35</td>
<td>0</td>
<td>0.5</td>
<td>20(100)</td>
<td>14</td>
<td>224</td>
<td>0</td>
<td>6.088</td>
</tr>
<tr>
<td>10(15000)</td>
<td>5</td>
<td>35</td>
<td>0</td>
<td>0.75</td>
<td>20(5000)</td>
<td>15</td>
<td>255</td>
<td>0</td>
<td>92.342</td>
</tr>
</tbody>
</table>

Table 5: Iterations (IT), function evaluations (F), backrackings (BK), and CPU time (T) for Newton-GMRES.

31
<table>
<thead>
<tr>
<th>Function(n)</th>
<th>IT</th>
<th>F</th>
<th>BK</th>
<th>T</th>
<th>Function(n)</th>
<th>IT</th>
<th>F</th>
<th>BK</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>1(1000)</td>
<td>5</td>
<td>56</td>
<td>0</td>
<td>0.9</td>
<td>11(500)</td>
<td>5</td>
<td>62</td>
<td>0</td>
<td>0.13</td>
</tr>
<tr>
<td>1(5000)</td>
<td>4</td>
<td>34</td>
<td>0</td>
<td>0.19</td>
<td>11(1000)</td>
<td>5</td>
<td>62</td>
<td>0</td>
<td>0.15</td>
</tr>
<tr>
<td>1(10000)</td>
<td>3</td>
<td>21</td>
<td>0</td>
<td>0.28</td>
<td>11(2000)</td>
<td>5</td>
<td>62</td>
<td>0</td>
<td>0.26</td>
</tr>
<tr>
<td>2(500)</td>
<td>4</td>
<td>40</td>
<td>0</td>
<td>0.07</td>
<td>12(100)</td>
<td>5</td>
<td>57</td>
<td>1</td>
<td>0.04</td>
</tr>
<tr>
<td>2(1000)</td>
<td>4</td>
<td>38</td>
<td>0</td>
<td>0.08</td>
<td>12(500)</td>
<td>5</td>
<td>61</td>
<td>1</td>
<td>0.18</td>
</tr>
<tr>
<td>2(2000)</td>
<td>3</td>
<td>21</td>
<td>0</td>
<td>0.08</td>
<td>12(1000)</td>
<td>5</td>
<td>57</td>
<td>1</td>
<td>0.18</td>
</tr>
<tr>
<td>3(50)</td>
<td>9</td>
<td>1737</td>
<td>10</td>
<td>0.24</td>
<td>13(100)</td>
<td>6</td>
<td>87</td>
<td>2</td>
<td>0.1</td>
</tr>
<tr>
<td>3(100)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>13(500)</td>
<td>6</td>
<td>85</td>
<td>2</td>
<td>0.3</td>
</tr>
<tr>
<td>3(200)</td>
<td>4</td>
<td>538</td>
<td>15</td>
<td>0.29</td>
<td>13(1000)</td>
<td>6</td>
<td>85</td>
<td>2</td>
<td>0.48</td>
</tr>
<tr>
<td>4(99)</td>
<td>8</td>
<td>182</td>
<td>5</td>
<td>0.07</td>
<td>14(100)</td>
<td>6</td>
<td>87</td>
<td>2</td>
<td>0.1</td>
</tr>
<tr>
<td>4(399)</td>
<td>8</td>
<td>261</td>
<td>20</td>
<td>0.19</td>
<td>14(500)</td>
<td>6</td>
<td>85</td>
<td>2</td>
<td>0.3</td>
</tr>
<tr>
<td>4(999)</td>
<td>8</td>
<td>261</td>
<td>20</td>
<td>0.6</td>
<td>14(1000)</td>
<td>6</td>
<td>85</td>
<td>2</td>
<td>0.48</td>
</tr>
<tr>
<td>5(1000)</td>
<td>7</td>
<td>255</td>
<td>0</td>
<td>0.4</td>
<td>15(500)</td>
<td>5</td>
<td>54</td>
<td>0</td>
<td>0.08</td>
</tr>
<tr>
<td>5(5000)</td>
<td>7</td>
<td>255</td>
<td>0</td>
<td>1.6</td>
<td>15(1000)</td>
<td>5</td>
<td>54</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>5(10000)</td>
<td>7</td>
<td>255</td>
<td>0</td>
<td>3</td>
<td>15(5000)</td>
<td>5</td>
<td>54</td>
<td>0</td>
<td>0.4</td>
</tr>
<tr>
<td>6(100)</td>
<td>5</td>
<td>50</td>
<td>0</td>
<td>0.14</td>
<td>16(1000)</td>
<td>4</td>
<td>34</td>
<td>0</td>
<td>0.005</td>
</tr>
<tr>
<td>6(500)</td>
<td>5</td>
<td>50</td>
<td>0</td>
<td>4.4</td>
<td>16(10000)</td>
<td>4</td>
<td>34</td>
<td>0</td>
<td>0.33</td>
</tr>
<tr>
<td>6(1000)</td>
<td>5</td>
<td>50</td>
<td>0</td>
<td>8.5</td>
<td>16(50000)</td>
<td>4</td>
<td>34</td>
<td>0</td>
<td>1.65</td>
</tr>
<tr>
<td>7(9)</td>
<td>5</td>
<td>90</td>
<td>11</td>
<td>0.04</td>
<td>17(100)</td>
<td>7</td>
<td>203</td>
<td>0</td>
<td>0.07</td>
</tr>
<tr>
<td>7(99)</td>
<td>5</td>
<td>90</td>
<td>12</td>
<td>0.13</td>
<td>17(500)</td>
<td>8</td>
<td>444</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>7(399)</td>
<td>4</td>
<td>114</td>
<td>2</td>
<td>1.05</td>
<td>17(1000)</td>
<td>8</td>
<td>528</td>
<td>0</td>
<td>0.8</td>
</tr>
<tr>
<td>8(1000)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>18(399)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>8(5000)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>18(999)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>8(10000)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>18(9999)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>9(2500)</td>
<td>16</td>
<td>737</td>
<td>5</td>
<td>1.25</td>
<td>19(100)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>9(5000)</td>
<td>14</td>
<td>551</td>
<td>2</td>
<td>2.25</td>
<td>19(500)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>9(10000)</td>
<td>13</td>
<td>436</td>
<td>0</td>
<td>3.6</td>
<td>19(1000)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>10(5000)</td>
<td>5</td>
<td>50</td>
<td>0</td>
<td>0.24</td>
<td>20(50)</td>
<td>14</td>
<td>329</td>
<td>0</td>
<td>3.455</td>
</tr>
<tr>
<td>10(10000)</td>
<td>5</td>
<td>50</td>
<td>0</td>
<td>0.46</td>
<td>20(100)</td>
<td>14</td>
<td>329</td>
<td>0</td>
<td>9.013</td>
</tr>
<tr>
<td>10(15000)</td>
<td>5</td>
<td>50</td>
<td>0</td>
<td>0.7</td>
<td>20(500)</td>
<td>15</td>
<td>375</td>
<td>0</td>
<td>137.7</td>
</tr>
</tbody>
</table>

Table 6: Iterations (IT), function evaluations (F), backtrackings (BK), and CPU time (T) for Newton-Bi-CGSTAB.
| Function(n) | IT | F  | BK | T   | Function(n) | IT | F  | BK | T   |
|------------|----|----|----|-----|------------|----|----|----|-----|------------|----|----|----|-----|
| 1(1000)    | 5  | 68 | 0  | 0.11 | 11(500)    | 5  | 68 | 0  | 0.15 |
| 1(5000)    | 4  | 40 | 0  | 0.24 | 11(1000)   | 5  | 66 | 0  | 0.18 |
| 1(10000)   | 4  | 40 | 0  | 0.47 | 11(2000)   | 5  | 76 | 0  | 0.35 |
| 2(500)     | 6  | 545| 0  | 1.25 | 12(100)    | 6  | 93 | 1  | 0.04 |
| 2(1000)    | 4  | 116| 0  | 0.5  | 12(500)    | 6  | 85 | 1  | 0.19 |
| 2(2000)    | 4  | 116| 0  | 0.9  | 12(1000)   | 6  | 85 | 1  | 0.23 |
| 3(50)      | 6  | 1713| 2 | 0.48 | 13(100)    | 5  | 65 | 1  | 0.08 |
| 3(100)     | 5  | 1220| 0 | 0.46 | 13(500)    | 5  | 70 | 2  | 0.3  |
| 3(200)     | 2  | 245| 0  | 0.23 | 13(1000)   | 5  | 74 | 2  | 0.6 |
| 4(99)      | 6  | 103| 2  | 0.05 | 14(100)    | 5  | 65 | 1  | 0.09 |
| 4(399)     | 6  | 101| 2  | 0.08 | 14(500)    | 6  | 99 | 2  | 0.35 |
| 4(999)     | 7  | 131| 2  | 0.29 | 14(1000)   | 5  | 70 | 2  | 0.49 |
| 5(1000)    | 6  | 75 | 0  | 0.08 | 15(500)    | 5  | 66 | 0  | 0.1 |
| 5(5000)    | 6  | 75 | 0  | 0.29 | 15(1000)   | 5  | 66 | 0  | 0.13 |
| 5(10000)   | 6  | 75 | 0  | 0.55 | 15(5000)   | 5  | 66 | 0  | 0.49 |
| 6(100)     | 5  | 58 | 0  | 0.13 | 16(1000)   | 4  | 34 | 0  | 0.04 |
| 6(500)     | 5  | 58 | 0  | 4.3  | 16(10000)  | 4  | 34 | 0  | 0.24 |
| 6(1000)    | 5  | 58 | 0  | 8.   | 16(50000)  | 4  | 34 | 0  | 1.2 |
| 7(9)       | 5  | 220| 3  | 0.12 | 17(100)    | 7  | 331| 0  | 0.15 |
| 7(99)      | 5  | 220| 3  | 0.36 | 17(500)    | 8  | 752| 0  | 1.4  |
| 7(399)     | 5  | 218| 2  | 1.15 | 17(1000)   | 8  | 780| 0  | 1.6  |
| 8(1000)    | *  | *  | *  | *   | 18(399)    | *  | *  | *  | *   |
| 8(5000)    | *  | *  | *  | *   | 18(999)    | *  | *  | *  | *   |
| 8(10000)   | *  | *  | *  | *   | 18(9999)   | *  | *  | *  | *   |
| 9(2500)    | 14 | 491| 4  | 1.1  | 19(100)    | *  | *  | *  | *   |
| 9(5000)    | 13 | 442| 4  | 2.   | 19(500)    | 10 | 175| 0  | 0.07 |
| 9(10000)   | 13 | 384| 2  | 3.1  | 19(1000)   | 10 | 175| 0  | 0.09 |
| 10(5000)   | 5  | 50 | 0  | 0.18 | 20(50)     | 14 | 329| 0  | 2.354 |
| 10(10000)  | 5  | 50 | 0  | 0.33 | 20(100)    | 14 | 329| 0  | 6.089 |
| 10(15000)  | 5  | 50 | 0  | 0.49 | 20(500)    | 15 | 375| 0  | 92.373 |

Table 7: Iterations (IT), function evaluations (F), backtrackings (BK), and CPU time (T) for Newton-TFQMR.
<table>
<thead>
<tr>
<th>Function(n)</th>
<th>IT</th>
<th>F</th>
<th>BK</th>
<th>T</th>
<th>Function(n)</th>
<th>IT</th>
<th>F</th>
<th>BK</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>1(1000)</td>
<td>11</td>
<td>23</td>
<td>0</td>
<td>0.05</td>
<td>11(500)</td>
<td>20</td>
<td>42</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>1(5000)</td>
<td>5</td>
<td>11</td>
<td>0</td>
<td>0.09</td>
<td>11(1000)</td>
<td>22</td>
<td>45</td>
<td>0</td>
<td>0.16</td>
</tr>
<tr>
<td>1(10000)</td>
<td>5</td>
<td>11</td>
<td>0</td>
<td>0.17</td>
<td>11(2000)</td>
<td>22</td>
<td>45</td>
<td>0</td>
<td>0.29</td>
</tr>
<tr>
<td>2(500)</td>
<td>12</td>
<td>27</td>
<td>1</td>
<td>0.05</td>
<td>12(100)</td>
<td>12</td>
<td>26</td>
<td>1</td>
<td>0.04</td>
</tr>
<tr>
<td>2(1000)</td>
<td>9</td>
<td>21</td>
<td>1</td>
<td>0.05</td>
<td>12(500)</td>
<td>11</td>
<td>24</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>2(2000)</td>
<td>7</td>
<td>18</td>
<td>1</td>
<td>0.07</td>
<td>12(1000)</td>
<td>11</td>
<td>24</td>
<td>1</td>
<td>0.19</td>
</tr>
<tr>
<td>3(50)</td>
<td>46</td>
<td>94</td>
<td>1</td>
<td>0.08</td>
<td>13(100)</td>
<td>15</td>
<td>33</td>
<td>2</td>
<td>0.13</td>
</tr>
<tr>
<td>3(100)</td>
<td>80</td>
<td>163</td>
<td>2</td>
<td>0.15</td>
<td>13(500)</td>
<td>14</td>
<td>31</td>
<td>2</td>
<td>0.35</td>
</tr>
<tr>
<td>3(200)</td>
<td>47</td>
<td>100</td>
<td>5</td>
<td>0.1</td>
<td>13(1000)</td>
<td>14</td>
<td>31</td>
<td>2</td>
<td>0.65</td>
</tr>
<tr>
<td>4(99)</td>
<td>120</td>
<td>287</td>
<td>45</td>
<td>0.5</td>
<td>14(100)</td>
<td>15</td>
<td>33</td>
<td>2</td>
<td>0.13</td>
</tr>
<tr>
<td>4(399)</td>
<td>146</td>
<td>358</td>
<td>59</td>
<td>1.1</td>
<td>14(500)</td>
<td>14</td>
<td>31</td>
<td>2</td>
<td>0.35</td>
</tr>
<tr>
<td>4(999)</td>
<td>128</td>
<td>319</td>
<td>56</td>
<td>2.9</td>
<td>14(1000)</td>
<td>14</td>
<td>31</td>
<td>2</td>
<td>0.65</td>
</tr>
<tr>
<td>5(1000)</td>
<td>35</td>
<td>74</td>
<td>2</td>
<td>0.12</td>
<td>15(500)</td>
<td>16</td>
<td>34</td>
<td>1</td>
<td>0.06</td>
</tr>
<tr>
<td>5(5000)</td>
<td>35</td>
<td>74</td>
<td>2</td>
<td>0.43</td>
<td>15(1000)</td>
<td>16</td>
<td>34</td>
<td>1</td>
<td>0.11</td>
</tr>
<tr>
<td>5(10000)</td>
<td>35</td>
<td>74</td>
<td>2</td>
<td>0.8</td>
<td>15(5000)</td>
<td>16</td>
<td>34</td>
<td>1</td>
<td>0.39</td>
</tr>
<tr>
<td>6(100)</td>
<td>8</td>
<td>17</td>
<td>0</td>
<td>0.16</td>
<td>16(1000)</td>
<td>6</td>
<td>13</td>
<td>0</td>
<td>0.02</td>
</tr>
<tr>
<td>6(500)</td>
<td>8</td>
<td>17</td>
<td>0</td>
<td>1.55</td>
<td>16(10000)</td>
<td>6</td>
<td>13</td>
<td>0</td>
<td>0.13</td>
</tr>
<tr>
<td>6(10000)</td>
<td>8</td>
<td>17</td>
<td>0</td>
<td>9</td>
<td>16(50000)</td>
<td>6</td>
<td>13</td>
<td>0</td>
<td>0.65</td>
</tr>
<tr>
<td>7(9)</td>
<td>12</td>
<td>37</td>
<td>4</td>
<td>0.04</td>
<td>17(100)</td>
<td>59</td>
<td>133</td>
<td>13</td>
<td>0.12</td>
</tr>
<tr>
<td>7(99)</td>
<td>12</td>
<td>37</td>
<td>4</td>
<td>0.15</td>
<td>17(500)</td>
<td>105</td>
<td>245</td>
<td>24</td>
<td>0.29</td>
</tr>
<tr>
<td>7(399)</td>
<td>12</td>
<td>37</td>
<td>4</td>
<td>0.48</td>
<td>17(1000)</td>
<td>125</td>
<td>305</td>
<td>32</td>
<td>0.6</td>
</tr>
<tr>
<td>8(1000)</td>
<td>6</td>
<td>13</td>
<td>0</td>
<td>0.04</td>
<td>18(399)</td>
<td>116</td>
<td>281</td>
<td>42</td>
<td>0.38</td>
</tr>
<tr>
<td>8(5000)</td>
<td>6</td>
<td>13</td>
<td>0</td>
<td>0.15</td>
<td>18(999)</td>
<td>114</td>
<td>278</td>
<td>44</td>
<td>0.65</td>
</tr>
<tr>
<td>8(10000)</td>
<td>6</td>
<td>13</td>
<td>0</td>
<td>0.28</td>
<td>18(9999)</td>
<td>80</td>
<td>198</td>
<td>31</td>
<td>4.3</td>
</tr>
<tr>
<td>9(2500)</td>
<td>19</td>
<td>42</td>
<td>1</td>
<td>0.38</td>
<td>19(100)</td>
<td>**</td>
<td>**</td>
<td>**</td>
<td>**</td>
</tr>
<tr>
<td>9(5000)</td>
<td>16</td>
<td>36</td>
<td>1</td>
<td>0.65</td>
<td>19(500)</td>
<td>13</td>
<td>28</td>
<td>1</td>
<td>0.04</td>
</tr>
<tr>
<td>9(10000)</td>
<td>17</td>
<td>39</td>
<td>1</td>
<td>1.25</td>
<td>19(1000)</td>
<td>12</td>
<td>26</td>
<td>1</td>
<td>0.051</td>
</tr>
<tr>
<td>10(5000)</td>
<td>6</td>
<td>13</td>
<td>0</td>
<td>0.09</td>
<td>20(50)</td>
<td>13</td>
<td>27</td>
<td>1</td>
<td>2.253</td>
</tr>
<tr>
<td>10(10000)</td>
<td>6</td>
<td>13</td>
<td>0</td>
<td>0.16</td>
<td>20(100)</td>
<td>8</td>
<td>17</td>
<td>1</td>
<td>3.856</td>
</tr>
<tr>
<td>10(15000)</td>
<td>6</td>
<td>13</td>
<td>0</td>
<td>0.24</td>
<td>20(500)</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>12.598</td>
</tr>
</tbody>
</table>

Table 8: Iterations (IT), function evaluations (F), backtracings (BK), and CPU time (T) for the SANE algorithm.
<table>
<thead>
<tr>
<th>Method</th>
<th>IT</th>
<th>F</th>
<th>T</th>
<th>FP(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newton-GMRES</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>8.33</td>
</tr>
<tr>
<td>Newton-Bi-CGSTAB</td>
<td>6</td>
<td>0</td>
<td>9</td>
<td>16.67</td>
</tr>
<tr>
<td>Newton-TFQMR</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>11.67</td>
</tr>
<tr>
<td>SANE</td>
<td>9</td>
<td>51</td>
<td>35</td>
<td>1.67</td>
</tr>
</tbody>
</table>

Table 9: Number of problems for which each method is a winner, and failure percentage (FP) for each method.

<table>
<thead>
<tr>
<th>Method</th>
<th>IT</th>
<th>F</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newton-GMRES</td>
<td>51</td>
<td>9</td>
<td>19</td>
</tr>
<tr>
<td>SANE</td>
<td>9</td>
<td>51</td>
<td>41</td>
</tr>
</tbody>
</table>

Table 10: SANE algorithm vs. Newton-GMRES.

References


