A multi-iterate method to solve systems of nonlinear equations [*]

M. Bierlaire F. Crittin M. Thémans

January 5, 2005

Report RO-060105
Operations Research Group ROSO
Institute of Mathematics
Ecole Polytechnique Fédérale de Lausanne

Abstract

We propose an extension of secant methods for nonlinear equations using a population of previous iterates. Contrarily to classical secant methods, where exact interpolation is used, we prefer a least squares approach to calibrate the linear model. We propose an explicit control of the numerical stability of the method.

We show that our approach can lead to an update formula. In that case, we prove the local convergence of the corresponding undamped quasi-Newton method. Finally, computational comparisons with classical quasi-Newton methods highlight a significant improvement in terms of robustness and number of function evaluations. We also present numerical tests showing the robust behavior of our method in the presence of noise.

1 Introduction

We consider the standard problem of identifying the solution of a system of nonlinear equations

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

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is a differentiable function. Since Newton, this problem has received a tremendous amount of attention. Newton's method and its many variations are still intensively analyzed and used in practice. The idea of Newton-like methods is to replace the nonlinear function F by a linear model, which approximates F in the neighborhood of the current iterate. The original Newton method invokes Taylor's theorem and uses the gradient matrix (the transpose of which is called the Jacobian) to construct the linear model. When the Jacobian is too expensive to evaluate, secant methods build the linear model

[*] This research is supported by SNF grants 205121-103598 and 205121-103598

based on the secant equation. Because secant methods exhibit a q-superlinear rate of convergence, they have been intensively analyzed in the literature.

The secant equation imposes that the linear model exactly matches the nonlinear function F at two successive iterates. If the number of unknowns n is strictly greater than 1, an infinite number of linear models verify the secant equation. Therefore, each secant method derives a specific update formula which arbitrarily picks one linear model among them. The most common strategies are called "least-change updates" and select the linear model which minimizes the difference between two successive models.

In this paper, we provide a class of algorithms generalizing these ideas. Instead of using only two successive iterates to determine this linear model, we maintain a "population" of previous iterates. This approach allows all the available information collected through the iterations to be explicitly used for calibrating the model.

An important feature of our method is that we do not impose an exact match between the model and the function. Instead, we use a least squares approach to request that the model fits the function "as well as possible". In this paper, we present the class of algorithms based on our method (Section 2.2) and prove that they are locally convergent (Section 3). This class of algorithms exhibits a faster convergence and a greater robustness than quasi-Newton methods for most numerical tests that we have performed (Section 4) at a cost of substantial linear algebra computation. Therefore it is valuable when the cost of evaluating F is high in comparison with the numerical algebra overhead.

2 Quasi-Newton methods

Quasi-Newton methods consider at each iteration the linear model

$$L_k(x; B_k) = F(x_k) + B_k(x - x_k) \tag{2}$$

which approximates $F(x)$ in the neighborhood of x_k and computes x_{k+1} as a solution of the linear system $L_k(x; B_k) = 0$. Consistently with most of the publications on this topic, quasi-Newton methods can be summarized as methods based on the following iterations:

$$x_{k+1} = x_k - B_k^{-1} F(x_k), \tag{3}$$

followed by the computation of B_{k+1}. The pure Newton method is obtained with $B_k = J(x_k) = \nabla F(x_k)^T$, the Jacobian of F evaluated at x_k, that is a $n \times n$ matrix such that entry (i, j) is $\partial F_i / \partial x_j$. We refer the reader to Dennis and Schnabel (1996) for an extensive analysis of Newton and quasi-Newton methods.

2.1 Secant methods

Broyden (1965) proposes a quasi-Newton method based on the *secant equations*, imposing the linear model L_{k+1} to exactly match the nonlinear function at iterates x_k and x_{k+1}, that is

$$\begin{aligned} L_{k+1}(x_k; B_{k+1}) &= F(x_k), \\ L_{k+1}(x_{k+1}, B_{k+1}) &= F(x_{k+1}). \end{aligned} \tag{4}$$

Subtracting these two equations and defining $y_k = F(x_{k+1}) - F(x_k)$ and $s_k = x_{k+1} - x_k$ we obtain the classical secant equation:

$$B_{k+1} s_k = y_k. \tag{5}$$

Clearly, if the dimension n is strictly greater than 1, there is an infinite number of matrices B_{k+1} satisfying (5). An arbitrary decision must consequently be made. The “least-change secant update” strategy, proposed by Broyden (1965), consists in selecting among the matrices verifying (5) the one minimizing variations (in Frobenius norm) between two successive matrices B_k and B_{k+1}. It leads to the following update formula

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)\, s_k^T}{s_k^T s_k}. \tag{6}$$

This method has been very successful, and has been widely adopted in the field. However, we believe that the idea of interpolating the linear model at only two iterates and ignoring previous iterates could be too restrictive. Therefore, we propose to use more than two iterates to build the linear model.

This idea has already been considered. Dennis and Schnabel (1996) say that “Perhaps the most obvious strategy is to require the model to interpolate $F(x)$ at other past points... One problem is that the directions tend to be linearly dependent or close to it, making the computation of (the approximation matrix) a poorly posed numerical problem”. Later, they write “In fact, multivariable generalizations of the secant method have been proposed ... but none of them seem robust enough for general use.”

There are few attempts to generalize this approach in the literature. A first generalization of the secant method is the *sequential secant method* proposed by Wolfe (1959) and discussed by Ortega and Rheinboldt (1970). The idea is to impose exact interpolation of the linear model on $n+1$ iterates instead of 2:

$$L_{k+1}(x_{k+1-j}; B_{k+1}) = F(x_{k+1-j}), \quad j = 0, 1, \ldots, n. \tag{7}$$

or, equivalently,

$$B_{k+1} s_{k-j} = y_{k-j}, \quad j = 0, 1, \ldots, n-1, \tag{8}$$

where $s_i = x_{k+1} - x_i$, and $y_i = F(x_{k+1}) - F(x_i)$, for all i. If the vectors $s_k, s_{k-1}, \ldots, s_{k-n+1}$ are linearly independent, there exists exactly one matrix B_{k+1} satisfying (8), which is

$$B_{k+1} = Y_{k+1} S_{k+1}^{-1} \tag{9}$$

where $Y_{k+1} = (y_k, y_{k-1}, \ldots, y_{k-n+1})$ and $S_{k+1} = (s_k, s_{k-1}, \ldots, s_{k-n+1})$.

Quoting Ortega and Rheinboldt (1970) "...(sequantial methods) are prone to unstable behavior and ... no satisfactory convergence results can be given". Nevertheless Gragg and Stewart (1976) propose a method which avoids instabilities by working with orthogonal factorizations of the involved matrices. Martinez (1979) gives three implementations of the idea proposed by Gragg and Stewart (1976) and some numerical experiments.

Multi-step quasi-Newton methods have been proposed by Moghrabi (1993), Ford and Moghrabi (1997) and Ford (1999) in the context of nonlinear programming. An interpolating path is built based on previous iterates, and used to produce an alternative secant equation. Interestingly, the best numerical results were obtained with no more than two steps.

We believe that the comments about the poor numerical stability of those methods found in major reference texts such as Dennis and Schnabel (1996) and Ortega and Rheinboldt (1970) have not encouraged researchers to pursue these investigatations. We provide here a successful multi-iterates appoach with robust convergence properties and exhibiting an excellent behavior on numerical examples. The idea of using a least squares approach is similar to an idea proposed in the physics litterature by Vanderbilt and Louie (1984), which has inspired other authors in the same field (Johnson, 1988, Eyert, 1996). Bierlaire and Crittin (forthcoming) have used a similar approach for solving noisy large scale transportation problems.

2.2 Population-based approach

We propose a class of methods calibrating a linear model based on several previous iterates. The difference with existing approaches is that we do not impose the linear model to interpolate the function. Instead, we prefer to identify the linear model which is as close as possible to the nonlinear function, in the least squares sense.

At each iteration, we maintain a finite population of previous iterates. Without loss of generality, we present the method assuming that all previous iterates $x_0, \ldots, x_{k+1}$ are considered. Our method belongs also to the quasi-Newton framework defined by (3), where B_{k+1} is computed as follows.

$$B_{k+1} = \operatorname*{argmin}_J \left(\sum_{i=0}^{k} \left\| \omega_{k+1}^i F(x_i) - \omega_{k+1}^i L_{k+1}(x_i; J) \right\|_2^2 + \left\| J\Gamma - B_{k+1}^0 \Gamma \right\|_F^2 \right) \quad (10)$$

where L_{k+1} is defined by (2) and $B_{k+1}^0 \in \mathbb{R}^{n \times n}$ is an a priori approximation of B_{k+1}. The role of the second term is to overcome the under-determination of the least squares problem based on the first term and also control the numerical stability of the method. The matrix Γ contains weights associated with the arbitrary term B_{k+1}^0, and the weights $\omega_{k+1}^i \in \mathbb{R}^+$ are associated with the previous iterates. Equation (10) can be written in matrix form as follows: $B_{k+1} =$

$$\operatorname*{argmin}_J \left\| J \begin{pmatrix} S_{k+1} & I_{n\times n} \end{pmatrix} \begin{pmatrix} \Omega & 0_{k\times n} \\ 0_{n\times k} & \Gamma \end{pmatrix} - \begin{pmatrix} Y_{k+1} & B^0_{k+1} \end{pmatrix} \begin{pmatrix} \Omega & 0 \\ 0 & \Gamma \end{pmatrix} \right\|_F^2$$

where $\Omega \in \mathbb{R}^{k+1}$ is a diagonal matrix with weights ω^i_{k+1} on the diagonal for $i = 0, \cdots, k$. The normal equations of this least squares problem lead to the following formula:

$$B_{k+1} = B^0_{k+1} + \left(Y_{k+1} - B^0_{k+1}S_{k+1}\right)\Omega^2 S^T_{k+1}\left(\Gamma\Gamma^T + S_{k+1}\Omega^2 S^T_{k+1}\right)^{-1}, \quad (11)$$

where $Y_{k+1} = (y_k, y_{k-1}, \ldots, y_0)$ and $S_{k+1} = (s_k, s_{k-1}, \ldots, s_0)$.

The role of the a priori matrix B^0_{k+1} is to overcome the possible underdetermination of problem (10). For example, choosing $B^0_{k+1} = B_k$ (similarly to classical Broyden-like methods) exhibits good properties. In that case, (11) becomes an update formula, and local convergence can be proved (see Section 3).

The weights ω^i_{k+1} capture the relative importance of each iterate in the population. Roughly speaking, they should be designed in the lines of the assumptions of Taylor's theorem, that is assigning more weight to points close to x_{k+1}, and less weight to points which are faraway. The matrix Γ captures the importance of the arbitrary terms defined by B^0_{k+1} for the identification of the linear model. The weights have to be finite, and Γ must be such that

$$\Gamma\Gamma^T + S_{k+1}\Omega^2 S^T_{k+1} \quad (12)$$

is safely positive definite. To ensure this property we describe below three possible approaches for choosing $\Gamma\Gamma^T$: the *geometrical approach*, based on specific geometric properties of the population, the *subspace decomposition* approach, decomposing $\mathbb{R}^n$ into the subspace spanned by the columns of S_{k+1} and its orthogonal complement, and the *numerical approach*, designed to guarantee a numerically safe positive definiteness of (12).

The *geometrical approach* assumes that $n + 1$ members of the population form a simplex, so that the columns of S_{k+1} span $\mathbb{R}^n$, and (12) is positive definite with $\Gamma\Gamma^T = 0$. In that case, (11) becomes

$$B_{k+1} = Y_{k+1}\Omega^2 S^T_{k+1}\left(S_{k+1}\Omega^2 S^T_{k+1}\right)^{-1}. \quad (13)$$

If there are exactly $n + 1$ iterates forming a simplex, the geometrical approach is equivalent to the interpolation method proposed by Wolfe (1959), and (13) is exactly (9), as S_{k+1} is square and non singular in that case. This approach have not shown good numerical behavior in practice as mentioned in Section 2. Also, it requires at least $n + 1$ iterates, and may not be appropriate for large-scale problems.

The *subspace decomposition* approach is based on the QR decomposition of S_{k+1}. We denote by r the rank of S_{k+1}, with $r \leq n$, and we have $S_{k+1} = QR$, where

$$Q = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \quad (14)$$

with Q_1 is $(n \times r)$, Q_2 is $(n \times n - r)$, and R is $(n \times k + 1)$. The r columns of Q_1 form an orthogonal basis of the range of S_{k+1}. We define now Γ such that

$$\Gamma = \begin{pmatrix} 0_{n \times r} & Q_2 \end{pmatrix} \tag{15}$$

that is Q where Q_1 has been replaced by a null matrix. With this construction $\Gamma\Gamma^T + S_{k+1}\Omega^2 S_{k+1}^T$ is invertible and $S_{k+1}\Gamma\Gamma^T = 0$. In the case where S_{k+1} spans the entire space then $r = n$, Γ is a null matrix and (11) is equivalent to (13).

With the subspace decomposition approach, the changes of F predicted by B_{k+1} in a direction orthogonal to the range of S_{k+1} is the same as the one predicted by the arbitrary matrix B^0_{k+1}. This idea is exactly the same as the one used by Broyden (1965) to construct his so called *Broyden's Good method*.

Numerical problems may happen when the columns of S_{k+1} are close to linear dependence. These are the problems already mentioned in the introduction, and reported namely by Ortega and Rheinboldt (1970) and Dennis and Schnabel (1996). Clearly, such problems do not occur when S_{k+1} has exactly one column, which leads to the classical Broyden method.

The *numerical approach* is designed to address both the problem of overcoming the under-determination, and of guaranteeing numerical stability. It is directly inspired by the modified Cholesky factorization proposed by Schnabel and Eskow (1991). The modified Cholesky factorization of a square matrix A creates a matrix E such that $A+E$ is safely positive definite, while computing its Cholesky factorization. It may namely happen that A has full rank, but with smallest eigenvalue very small with regard to machine precision. In that case, E is non zero despite the fact that A is non singular. We apply this technique with $A = S_{k+1}\Omega^2 S_{k+1}^T$ and $E = \Gamma\Gamma^T$. So, if the matrix $S_{k+1}\Omega^2 S_{k+1}^T$ is safely positive definite, $\Gamma\Gamma^T = 0$ and (11) reduces to (13). If not, the modified Cholesky factorization guarantees that the role of the arbitrary term Γ is minimal.

We now emphasize important advantages of our generalization combined with the *numerical approach*. Firstly, contrarily to interpolation methods, our least squares model allows to use more than p points to identify a model in a subspace of dimension p (where $p \leq n$). This is very important when the objective function is expensive to evaluate. Indeed, we make an efficient use of all the available information about the function to calibrate the secant model. It is namely advantageous compared to Broyden's method, where only two iterates are explicitly used to build the model, while previous iterates only play an implicit role due to the "least-change" principle. Secondly, the numerical approach proposed above controls the numerical stability of the model construction process, when a sequence of iterates may be linearly dependent. Finally, the fact that existing methods are special cases of our approach allows to generalize the theoretical and practical properties already published in the literature, and simplifies their extension to our context. We apply this principle to the local convergence analysis in section 3. The main drawback is the increase in numerical linear algebra as the least squares problem (10) must be solved at each iteration. Therefore, it is particularly appropriate for problems where F is very expensive to compute.

We conclude this section by showing that our population-based update formula is a generalization of Broyden update. Actually, the classical Broyden update (6) is a special case of our update formula (11), if $B^0_{k+1} = B_k$, the population contains just two iterates x_k and x_{k+1}, and the *subspace decomposition* approach is used. The secant equation (5) completely defines the linear model in the one-dimensional subspace spanned by $s_k = x_{k+1} - x_k$, while an arbitrary decision is made for the rest of the model. If we define $\omega^k_{k+1} = 1$ and Γ is given by (15) with $r = 1$, we can write (11) as

$$B_{k+1} = B_k + (y_k - B_k s_k)\, s_k^T \left(\Gamma\Gamma^T + s_k s_k^T\right)^{-1}. \tag{16}$$

The equivalence with (6) is due to the following equality

$$s_k^T \left(\Gamma\Gamma^T + s_k s_k^T\right)^{-1} = s_k^T \frac{1}{s_k^T s_k}, \tag{17}$$

obtained from the fact that $s_k^T \Gamma\Gamma^T = 0$, by (15).

3 Local convergence analysis

We show that if $\Gamma\Gamma^T$ is determined by the numerical approach described in Section 2.2, then the undamped algorithm described in Section 3.1, where B_{k+1} is defined by (11) in its update form (*i.e.* $B^0_{k+1} = B_k$), locally converges to a solution of (1) if the following assumptions are verified. Note that the assumptions made on the problem are similar to those given by Broyden (1965).

Assumptions on the problem:

(P1) $F : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable in an open convex set $\mathcal{D}$.

(P2) The system of equations has a solution, that is $\exists\, x^* \in \mathcal{D}$ such that $F(x^*) = 0$.

(P3) $J(x)$ is Lipschitz continuous at x^* with constant K_{lip}, that is

$$\|J(x) - J(x^*)\| \leq K_{\text{lip}} \|x - x^*\| \quad \forall x \in \mathcal{D}. \tag{18}$$

in the neighborhood $\mathcal{D}$.

(P4) $J(x^*)$ is non-singular and there exists $\gamma > 0$ such that $\|J(x^*)^{-1}\| < \gamma$.

Assumptions on the algorithm:

(A1) The algorithm is based on the iteration (3) with x_0 and B_0 as initial guess.

(A2) B_k is generated by (11) with $B^0_{k+1} = B_k$.

(A3) $\Gamma\Gamma^T$ is computed using the *numerical approach.*

(A4) $\forall i \leq k$, we have $\omega_{k+1}^i \leq M_\omega$ for all k and some constant $M_\omega > 0$.

(A5) The size of the population $\mathcal{P}$ is bounded above by $M_\mathcal{P}$ where $M_\mathcal{P} > 0$ is a constant.

The notation $\|\cdot\|$ is used for the l_2 vector norm $\|x\| = (x^T x)^{\frac{1}{2}}$ as well as for the Frobenius matrix norm $\|A\|$. The notation $\|\cdot\|_2$ is used for the l_2 matrix norm $\|A\|_2$. For the sake of simplification, we denote $\omega_{k+1}^i = \omega_i$, $S = S_{k+1}$, $Y = Y_{k+1}$ and $I_p = \{0, 1, \ldots, p\}$. The proof uses some lemmas. Lemma 1 and 2 are classical results from the literature. Lemmas 3–5 are technical lemmas related to our method. Their proofs are provided in the appendix.

Lemma 1 *Let* $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ *be continuously differentiable in the open convex* $D \subset \mathbb{R}^n$, $x \in D$, *and let* J *be Lipschitz continuous at* x *in the neighborhood* D *with constant* K_{lip}. *Then for any* $u, v \in D$,

$$\|F(v) - F(u) - J(x)(v - u)\| \leq K_{lip} \frac{\|v - x\| + \|u - x\|}{2} \|v - u\|. \qquad (19)$$

Proof. See, for example, Dennis and Schnabel, 1996. □

Lemma 2 *Let* $A, C \in \mathbb{R}^{n \times n}$ *and assume that* A *is invertible, with* $\|A^{-1}\| \leq \mu$. *If* $\|A - C\| \leq \beta$ *and* $\beta\mu < 1$, *then* C *is also invertible and*

$$\left\|C^{-1}\right\| \leq \frac{\mu}{1 - \beta\mu}. \qquad (20)$$

Proof. This lemma is known as the Banach Perturbation Lemma. (See, for example, Ortega and Rheinboldt, 1970). □

Lemma 3 *If assumptions (A4)-(A5) are verified, then*

$$\begin{aligned} \|S\Omega^2 S^T\| &\leq 2M_\mathcal{P} M_\omega^2 \max_{i \in I_{k+1}} \|x_i - x^*\|^2, & (21) \\ \|\Omega^2 S^T\| &\leq \sqrt{2M_\mathcal{P}} M_\omega^2 \max_{i \in I_{k+1}} \|x_i - x^*\|. & (22) \end{aligned}$$

where x^* *is solution of (1).*

Lemma 4 *If assumptions (P1),(P2) and (P3) are verified then:*

$$\|(Y - J(x^*)S)\| \leq \sqrt{2M_\mathcal{P}} K_{lip} \max_{i \in I_{k+1}} \left(\|x_i - x^*\|^2\right) \qquad (23)$$

where x^* *is solution of (1).*

Lemma 5 *If assumption (A3) is verified, then*

$$\left\| \left(\Gamma\Gamma^T + S\Omega^2 S^T \right)^{-1} \right\|_2 \leq \frac{1}{\tau} \tag{24}$$

where $\tau > 0$.

The parameter τ in Lemma 5 controls the way we perturb $S\Omega^2 S^T$. It guarantees that the smallest eigenvalue of $\left(\Gamma\Gamma^T + S\Omega^2 S^T\right)$ is strictly greater than τ and, therefore, safely positive in a finite arithmetic context if τ is properly chosen. Schnabel and Eskow (1991) suggest to choose $\tau = (\text{macheps})^{\frac{1}{3}}$ where macheps is the machine epsilon.

Theorem 6 *Let assumptions (P1) to (P4) hold for the problem and assumptions (A1) to (A5) hold for the algorithm. Then there exists two non-negative constants* α_1 *and* α_2 *such that for each* x_k *and* B_k*:*

$$\begin{aligned} \|B_{k+1} - J(x^*)\| \quad \leq & \quad \left(1 + \alpha_1 \max_{i \in I_{k+1}} \|x_i - x^*\|^2\right) \|B_k - J(x^*)\| \\ + & \quad \alpha_2 \max_{i \in I_{k+1}} \|x_i - x^*\|^3 . \end{aligned} \tag{25}$$

Proof. From the update formula (11), and defining

$$\begin{aligned} T_1 &= I - S\Omega^2 S^T (\Gamma\Gamma^T + S\Omega^2 S^T)^{-1} \\ T_2 &= (Y - J(x^*)S)\Omega^2 S^T (\Gamma\Gamma^T + S\Omega^2 S^T)^{-1}, \end{aligned}$$

we obtain

$$\begin{aligned} \|B_{k+1} - J(x^*)\| &= \|B_k - J(x^*) + [(J(x^*)S - J(x^*)S) + (Y - B_k S)]\,\Omega^2 S^T (\Gamma\Gamma^T + S\Omega^2 S^T)^{-1}\| \\ &\leq \|T_1\| \|B_k - J(x^*)\| + \|T_2\|. \end{aligned}$$

From Lemmas 3 and 5 we obtain

$$\begin{aligned} \|T_1\| &\leq \|I\| + \|S\Omega^2 S^T\| \|(\Gamma\Gamma^T + S\Omega^2 S^T)^{-1}\| & (26) \\ &\leq 1 + \alpha_1 \max_{i \in I_{k+1}} \|x_i - x^*\|^2, & (27) \end{aligned}$$

with

$$\alpha_1 = \frac{2\sqrt{n}}{\tau} M_{\mathcal{P}} M_\omega^2 > 0.$$

We conclude the proof using Lemmas 3, 4 and 5 to show that:

$$\begin{aligned} \|T_2\| &\leq \|(Y - J(x^*)S)\| \|\Omega^2 S^T\| \|(\Gamma\Gamma^T + S\Omega^2 S^T)^{-1}\| & (28) \\ &\leq \alpha_2 \max_{i \in I_{k+1}} \|x_i - x^*\|^3, & (29) \end{aligned}$$

with

$$\alpha_2 = \frac{2\sqrt{n}}{\tau} K_{lip} M_{\mathcal{P}} M_\omega^2 > 0.$$

□

Theorem 7 *Let assumptions (P1) to (P3) hold for the problem and assumptions (A1) to (A5) hold for the algorithm. Then for each* $r \in]0,1[$*, there exists* $\varepsilon(r)$ *and* $\delta(r)$ *such that for*

$$\|x_0 - x^*\| \leq \varepsilon(r) \tag{30}$$

and

$$\|B_0 - J(x^*)\| \leq \delta(r) \tag{31}$$

the sequence $x_{k+1} = x_k - B_k^{-1}F(x_k)$ *is well defined and converges* q*-linearly to* x^* *with* q*-factor at most* r*. Furthermore, the sequences* $\{\|B_k\|\}_k$ *and* $\{\|B_k^{-1}\|\}_k$ *are uniformly bounded.*

Proof. The structure of the demonstration is similar to the proof of Theorem 3.2 in Broyden et al. (1973). We have purposedly skipped some identical technical details.

First choose $\varepsilon(r) = \varepsilon$ and $\delta(r) = \delta$ such that

$$\gamma(1+r)\left(K_{lip}\varepsilon + 2\delta\right) \leq r \tag{32}$$

and

$$\left(2\alpha_1 + \alpha_2 \frac{\varepsilon}{1-r}\right) \frac{\varepsilon^2}{1-r^2} \leq \delta. \tag{33}$$

We invoke Lemma 2 with $\mu = \gamma$ and $\beta = 2\delta$ to prove that B_0 is non-singular and

$$\|B_0^{-1}\| < \gamma(1+r). \tag{34}$$

Note that assumption $2\delta\gamma < 1$ for Lemma 2 is directly deduced from (32).

The improvement after the first iteration, that is

$$\|x_1 - x^*\| \leq r\|x_0 - x^*\| \tag{35}$$

is independent of the specific update formula and, therefore, is proven in Broyden et al. (1973).

The result for iteration k is proven with an induction argument based on the following recurrence assumptions:

$$\begin{aligned} \|B_m - J^*\| &\leq 2\delta & (36)\\ \|x_{m+1} - x^*\| &\leq r\|x_m - x^*\| & (37) \end{aligned}$$

for all $m = 1, \ldots, k-1$.

We first prove that $\|B_k - J^*\| \leq 2\delta$ using Theorem 6. From (25) we deduce $\|B_{m+1} - J(x^*)\| - \|B_m - J(x^*)\|$

$$\begin{aligned} &\leq \alpha_1 \max_{i \in I_{m+1}} \|x_i - x^*\|^2 \|B_m - J(x^*)\| + \alpha_2 \max_{i \in I_{m+1}} \|x_i - x^*\|^3 \\ &\leq \alpha_1 r^{2(m+1)} \varepsilon^2 2\delta + \alpha_2 r^{3(m+1)} \varepsilon^3. \end{aligned} \tag{38}$$

Summing both sides of (38) for m ranging from 0 to $k-1$, we deduce that

$$\begin{aligned} \|B_k - J(x^*)\| &\le \|B_0 - J(x^*)\| + \left(2\alpha_1\delta + \alpha_2\frac{\varepsilon}{1-r}\right)\frac{\varepsilon^2}{1-r^2} && (39) \\ &\le 2\delta, && (40) \end{aligned}$$

where (40) derives from (31) and (33).

The fact that B_k is invertible and $\|B_k^{-1}\| \le \gamma(1+r)$ is again a direct application of the Banach Perturbation Lemma 2. Following again Broyden et al. (1973), we can now obtain (37) for $m = k$, concluding the induction proof. □

3.1 Undamped and damped quasi-Newton methods

All the algorithms presented in Section 2.1 and 2.2 are based on the following structure.

- Given $F : \mathbb{R}^n \to \mathbb{R}^n$, $x_0 \in \mathbb{R}^n$ and $B_0 \in \mathbb{R}^{n\times n}$
- While stopping criteria is not verified:
  - Find s solving $B_k s = -F(x_k)$,
  - Evaluate $F(x_{k+1})$ where $x_{k+1} = x_k + s$,
  - Compute B_{k+1}.

This general algorithm is often called *undamped* quasi-Newton method, *i.e.* without any step control or globalization methods. It allows to compare different type of algorithms, in term of number of function evaluations, and their robustness without introducing a bias due to the step control or the globalization method. Consequently, the algorithms differ only by the method used to compute B_{k+1}.

The main drawback of undamped methods is that we cannot ensure convergence from remote starting points. Moreover, Newton-like methods without any control on the step lengths may encounter several other sources of failure. For instance, the components of the unknown vector (x) or the function vector (F) or the Jacobian approximate (B_k) may become arbitrarily large.

Globalization strategies can be grouped into two distinct frameworks: linesearch and trust-region. Linesearch approaches are applied to a merit function based on F, used to measure progress toward a solution of $F(x) = 0$ (see for instance Nocedal and Wright, 1999). Trust-region methods and filter-trust-region methods (see Gould et al., 2005) can be used to solve the associated nonlinear least squares problem:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|F(x)\|_2^2 \tag{41}$$

The main disadvantage of the second type of globalization is that the iterates can be stucked in a local minimum of (41), which is not a solution of $F(x) = 0$.

As we want to keep solving the original problem $F(x) = 0$, we adopt in this paper the linesearch approach.

When integrating a linesearch strategy to the previous undamped quasi-Newton framework, we obtain the following structure.

- Given $F : \mathbb{R}^n \to \mathbb{R}^n$, $x_0 \in \mathbb{R}^n$ and $B_0 \in \mathbb{R}^{n \times n}$
- While stopping criteria is not verified:
  - Find s solving $B_k s = -F(x_k)$;
  - Determine a step length $\alpha_k > 0$;
  - Evaluate $F(x_{k+1})$ where $x_{k+1} = x_k + \alpha_k s$;
  - Compute B_{k+1}.

This general method is called *damped* quasi-Newton method. In the following, we describe how we determine the step α_k at each iteration of the algorithm using the classical sum-of-squares merit function

$$m(x_k) = \frac{1}{2}\|F(x_k)\|_2^2 = \frac{1}{2}\sum_{i=1}^{n} F_i^2(x_k)$$

to measure progress toward a solution of the system F. We choose a step α_k satisfying the following Armijo-type condition with $\beta \in (0, 1)$:

$$m(x_k + \alpha_k s) \leq m(x_k) + \alpha_k \beta \nabla m(x_k)^T s. \tag{42}$$

Note that β is a parameter which defines the quality of the decrease we want to obtain. Condition (42) is valid only if the quasi-Newton direction s is a descent direction for m in x_k, that is:

$$\nabla m(x_k)^T s < 0. \tag{43}$$

If condition (43) holds, we find a step α_k satisfying (42) using a backtracking strategy. Unfortunately, we do not have the guarantee that our quasi-Newton direction $s = -B_k^{-1} F(x_k)$ is a descent direction for m, unless B_k is close enough to the real Jacobian at x_k, $J(x_k) = \nabla F(x_k)^T$, and $\nabla m(x_k)^T s$ is bounded below. Consequently, we use the following sequential procedure to find a descent direction for the merit function in the current iterate x_k:

- Check whether the quasi-Newton direction $s = -B_k^{-1} F(x_k)$ is a descent direction for m in x_k;
- If not, compute using the modified Cholesky factorization (see Schnabel and Eskow, 1999) an auxiliary direction $\bar{s}$

  $$-(B_k^T B_k + \tau I)^{-1} B_k^T F(x_k)$$

  where $\tau > 0$ and I is the identity matrix in dimension n. According to Nocedal and Wright (1999), we can always choose τ to ensure that $\nabla m(x_k)^T s$ is bounded below.

- Check whether the quasi-Newton direction $\bar{s}$ is a descent direction for m in x_k;
- If not, do the following:
  - Update the current approximation of the Jacobian B_k with a new point close to x_k to get B_k^+. More precisely, we take a step of length $1e-4$ in the direction s. The goal is to try to get a good local approximation of $J(x_k)$;
  - Compute the direction $s^+ = -(B_k^+)^{-1}F(x_k)$;

  and restart the process with s^+.

Note that we compute the directional derivative of the merit function m in a direction s, $\nabla m(x)^T s$, using a finite differences procedure.

4 Numerical Results

4.1 General behavior

We present here an analysis of the performance of our method, in comparison to classical algorithms. All algorithms and test functions have been implemented with the package Octave (Eaton, 1997) and computations have been done on a desktop equipped with 3GHz CPU in double precision. The machine epsilon is about 2.2e-16.

The numerical experiments were carried out on a set of 43 test functions. For 37 of them, we consider five instances of dimension $n = 6, 10, 20, 50, 100$. We obtain a total of 191 problems. This set is composed of the four standard nonlinear systems of equations proposed by Dennis and Schnabel (1996) (that is, *Extended Rosenbrock Function*, *Extended Powell Singular Function*, *Trigonometric Function*, *Helical Valley Function*), three functions from Broyden (1965), five functions proposed by Kelley (2003) in his book on Newton's method (that is, *Arctangent Function*, *a Simple Two-dimensional Function*, *Chandrasekhar H-equation*, *Ornstein -Zernike Equations*, *Right Preconditioned Convection-Diffusion Equation*), three linear systems of equations (see Appendix), the test functions given by Spedicato and Huang (1997) and some test functions of the collection proposed by Moré et al. (1981). For each problem, we have used the starting point proposed in the original paper. Note that the results include all these problems.

The algorithms are based on both the damped and undamped quasi-Newton framework given in Section 3.1 with the following characteristics: the initial Jacobian approximation B_0 is the same for all algorithms and equal to the identity matrix. The stopping criterion is a composition of three conditions: small residual, that is $\|F(x_k)\|/\|F(x_0)\| \leq 10e{-}6$, maximum number of iterations ($k \geq 200$ for problems of size $n \leq 20$ and $k \geq 500$ for problems of size $n > 20$), and divergence, diagnosed if $\|F(x_k)\| \geq 10e10$ or if a descent direction has not been found after several updates of the approximate Jacobian in the linesearch procedure (meaning that we have not been able to find a sufficiently good approximation of the Jacobian).

We consider four quasi-Newton methods:

1. Broyden's Good Method (BGM), using the update (6).

2. Broyden's Bad Method (BBM), also proposed by Broyden (1965). It is based on the following secant equation:

$$s_k = B_{k+1}^{-1} y_k. \tag{44}$$

   and directly computes the inverse of B_k:

$$B_{k+1}^{-1} = B_k^{-1} + \frac{\left(s_k - B_k^{-1} y_k\right) y_k^T}{y_k^T y_k}. \tag{45}$$

   Broyden (1965) describes this method as "bad", that is numerically unstable. However, we have decided to include it in our tests for the sake of completeness. Moreover, as discussed below, it does not always deserve its name.

3. The Hybrid Method (HMM) proposed by Martinez (1982). At each iteration, the algorithm decides to apply either BGM or BBM. Martinez (2000) observes a systematic improvement of the Hybrid approach with respect to each individual approach. As discussed below, we reach similar conclusions.

4. Our population-based approach, called Generalized Secant Method (GSM), defined by (11) in its update form with $B_{k+1}^0 = B_k$ using the *numerical approach* described in Section 2.2, with $\tau = (\text{macheps})^{\frac{1}{3}}$ and a maximum of $p = \max(n, 10)$ previous iterates in the population. Indeed, including all previous iterates, as proposed in the theoretical analysis, may generate memory management problems, and anyway does not significantly affect the behavior of the algorithm. The weights are defined as

$$\omega_{k+1}^i = \frac{1}{\|x_{k+1} - x_i\|^2} \qquad \forall i \in I_p \tag{46}$$

The measure of performance is the number of function evaluations to reach convergence. Indeed we are interested in applying the method on computationnally expensive systems, where the running time is dominated by the function evaluations. We are presenting the results following the performance profiles analysis method proposed by Dolan and Moré (2002).

If $f_{p,a}$ is the performance index (the number of function evaluations in our case) of algorithm a on problem p, then the *performance ratio* is defined by

$$r_{p,a} = \frac{f_{p,a}}{\min_a\{f_{p,a}\}}, \tag{47}$$

if algorithm a has converged for problem p, and $r_{p,a} = r_{\mathrm{fail}}$ otherwise, where r_{fail} must be strictly larger than any performance ratio (47). For any given threshold π, the overall performance of algorithm a is given by

$$\rho_a(\pi) = \frac{1}{n_p}\Phi_a(\pi) \tag{48}$$

where n_p is the number of problems considered, and $\Phi_a(\pi)$ is the number of problems for which $r_{p,a} \leq \pi$.

In particular, the value $\rho_a(1)$ gives the probability that algorithm a wins over all other algorithms. The value $\lim_{\pi \to r_{\mathrm{fail}}} \rho_a(\pi)$ gives the probability that algorithm a solves a problem and, consequently, provides a measure of the robustness of each method.

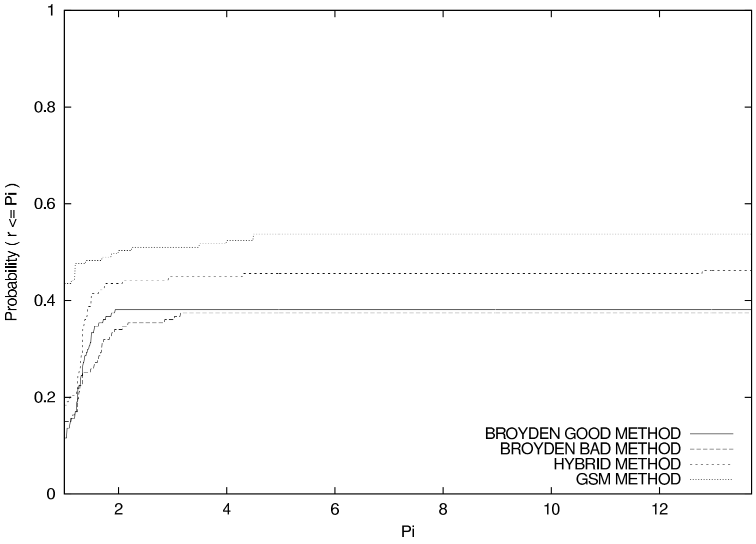


Figure 1: Performance Profile

We first analyze the performance profile of all algorithms described above without globalization strategy on all problems. The performance profile is reported on Figure 1. A zoom on π between 1 and 5 is provided in Figure 2.

The results are very satisfactory for our method. Indeed, we observe that GSM is the most efficient and the most robust algorithm among the challenged quasi-Newton methods.

We also confirm results by Martinez (2000) showing that the Hybrid method is more reliable than BGM and BBM. Indeed, it converges on almost 50% of the problems, while each Broyden method converges only on less than 40% of the cases. Moreover, HMM wins more often than BGM and BBM does, and is also more robust, as its performance profile grows faster than the profile for BGM and BBM. The relative robustness of BGM and BBM is comparable.

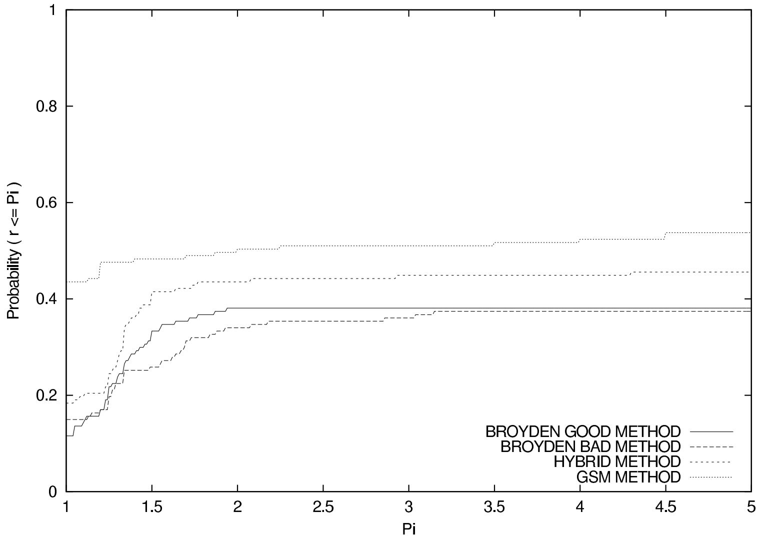


Figure 2: Performance Profile on (1,5)

Even if GSM is the most reliable algorithm, note that it only converges on 55% of the 191 runs. We now present the performance profile for all algorithms in their damped version, that is making use of the linesearch strategy presented in Section 3.1, on Figure 3. A zoom for π between 1 and 3 is provided in Figure 4. Firstly we observe that the globalization technique significantly improves the robustness of all four presented algorithms as expected. Secondly and most importantly, GSM remains the best algorithm in terms of efficiency and robustness. More precisely, GSM is the best algorithm on more than 60% of the problems and is able to solve more than 80% of the 191 considered problems. From Figure 4, we note also that when GSM is not the best method, it converges within a factor of 2 of the best algorithm for most problems.

The performance profile analysis depends on the number of methods that are being compared. Therefore, we like to present a comparison between BGM and GSM only, as BGM is probably the most widely used method. The significant improvement provided by our method over Broyden's method is illustrated by Figure 5 considering the undamped version of both algorithms. Figure 6 shows the superiority of GSM as well, when both algorithms are globalized using the linesearch strategy.

In this paper, in the context of solving systems of nonlinear equations, we focused on quasi-Newton methods which do not use information about the derivative of the system to be solved. We have already shown that GSM is a very competitive derivative-free algorithm. To conclude our numerical experiments, we like to compare our method with an algorithm using derivative information.

We consider a method belonging to the family of inexact Newton methods

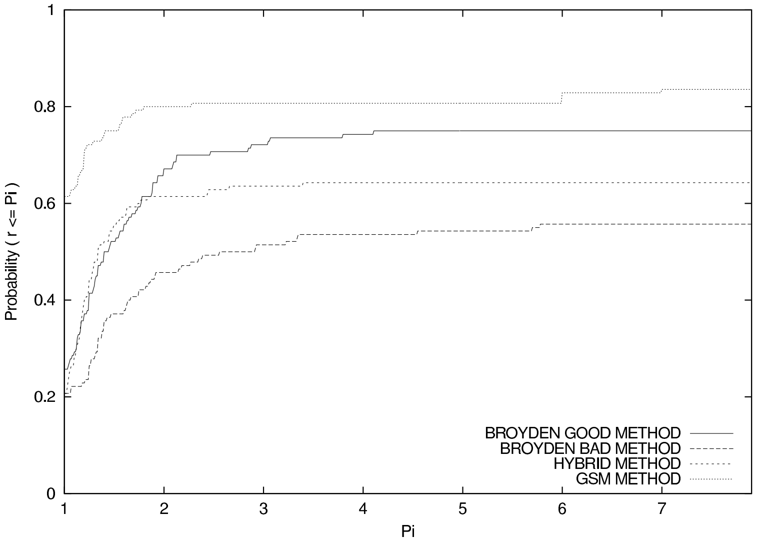


Figure 3: Performance Profile with linesearch

which identify a direction d_k satisfying the inexact Newton condition:

$$\|F(x_k) + J(x_k)d_k\| \leq \eta_k \|F(x_k)\| \tag{49}$$

for some $\eta_k \in [0, 1)$. The most conventional inexact Newton method uses iterative techniques to compute the Newton step d_k using (49) as a stopping criterion. Among these iteratives techniques, Krylov-based linear solvers are generally chosen. Newton-Krylov methods need to estimate Jacobian-vector products using finite differences approximations in the appropriate Krylov subspace.

We now challenge GSM against the Newton-Krylov method presented by Kelley (2003). The considered version of this method uses the iterative linear GMRES (proposed by Saad and Schultz, 1986) and a parabolic linesearch via three interpolation points. Similarly to the Newton-Krylov algorithm, we allow GSM to use a finite differences approximation of the initial Jacobian. From Figure 7, we observe that GSM is competitive with Newton-Krylov both in terms of efficiency and robustness. This result is very satisfactory as Newton-Krylov methods have been proven to be very efficient methods to solve systems of nonlinear equations.

4.2 Behavior in presence of noise

In practice the evaluation of systems of nonlinear equations often returns a result that is affected by noise, in particular if the evaluation is the outcome of simulator runs. For example Bierlaire and Crittin (forthcoming) describe such a problem in the context of transportation applications. Therefore, we

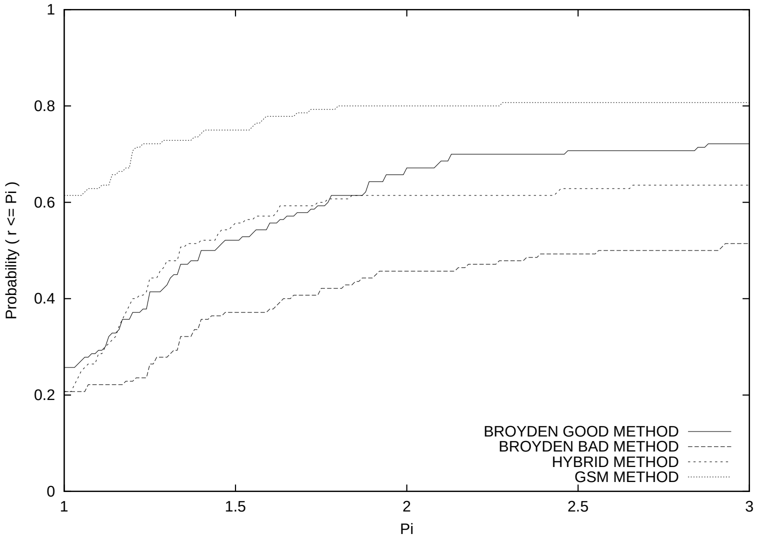


Figure 4: Performance Profile on (1,3) with linesearch

conclude this section by an empirical analysis of the behavior of our method in the presence of noise in the function. Indeed, we speculate that the use of a larger sample of iterates within a least squares framework smooths the impact of noise on the method.

We consider a random function described by:

$$G(x) = F_s(x) + \phi(x) \tag{50}$$

where $F_s : \mathbb{R}^n \to \mathbb{R}^n$ is deterministic and $\phi(x)$ is a random perturbation. We want to identify x such that $F_s(x) = 0$, but we are not able to compute $F_s(x)$ accurately.

We consider two types of random noise:

1. Similarly to Choi and Kelley (2000), we first assume that the noise decreases near the solution, more precisely:

   $$\phi(x) \sim N(0, \alpha^2 \|x - x^*\|^2) \text{ and } G(x_0) = F_s(x_0) = 0. \tag{51}$$

   In this case, the noise is named *proportional.*

2. We then assume that the noise is constant, more precisely:

   $$\phi(x) \sim N(0, \alpha^2). \tag{52}$$

   In this case, the noise is named *absolute.*

We have selected two problems where the behavior of BGM and GSM in their undamped version are almost similar in the deterministic case. Please

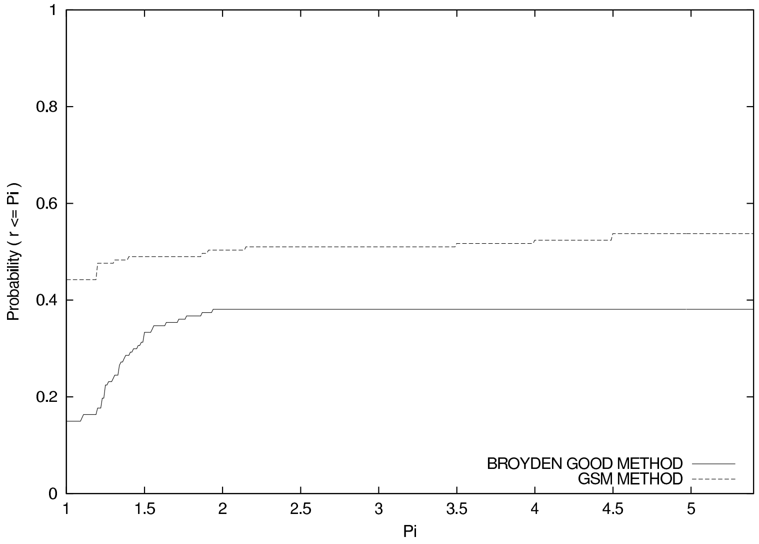


Figure 5: Performance profile – Broyden's Good Method and GSM –

note that we do not perform tests using the damped quasi-Newton framework as the underlying globalization strategy makes use of finite differences, which is not compatible with the stochasticity present in the problems considered in this subsection. For each function and each type of noise the results are presented for 4 levels of stochasticity, *i.e.* for four different values of the parameter α defined in equations (51) and (52). We plot the relative nonlinear residual, that is $\|G(x_k)\|/\|G(x_0)\|$, against the number of function evaluations.

First we consider a problem given by Spedicato and Huang (1997) and fully described in Section 6.4 in the Appendix. The results obtained with the proportional noise are presented in Figure 8. Figure 8(a) illustrates the deterministic case, with $\phi(x) = 0$, where BGM is slightly better than GSM. When a noise with small variance ($\alpha = 0.001$, Figure 8(b)) is present, GSM decreases the value of the residual pretty quickly, while the descent rate of BGM is much slower. When the variance of the noise increases ($\alpha = 0.01$ in Figure 8(c), and $\alpha = 1$ in Figure 8(d)), the BGM is trapped in higher values of the residual, while GSM achieves a significant decrease. The results obtained with the absolute noise are presented in Figure 9. The values of α are the same as above. The behavior of the two methods is almost the same as for the proportional noise. GSM reaches a lower level than BGM of the residual for small ($\alpha = 0.001$, Figure 9(b)) and medium ($\alpha = 0.01$, Figure 9(c)) variances. When the variance is higher ($\alpha = 1$, Figure 9(d)) none of the two methods is able to significantly decrease the relative residual.

The same tests have been accomplished with the Extended Rosenbrock Function given by Dennis and Schnabel (1996) and fully described in Section 6.5 in the Appendix. Figure 10 reports the behavior of GSM and BGM applied to

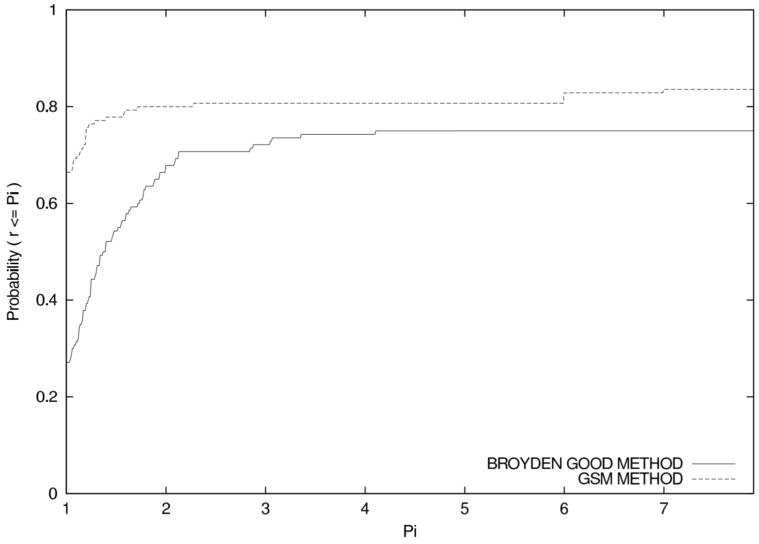


Figure 6: Performance profile with linesearch – Broyden's Good Method and GSM –

this problem perturbated with a proportional noise. Figure 10(a) reports the relative residual of the smooth system ($\alpha = 0$). In the presence of the small noise ($\alpha = 0.0001$, Figure 10(b)) both methods converge but BGM needs more than twice the number of iterations needed by GSM. When the noise increases ($\alpha = 0.01$, Figure 10(c)) BGM is totally disrupted and diverges, while GSM still converges in less than 20 iterations. With the higher value of the noise ($\alpha = 1$, Figure 10(c)) both methods are stalled, but GSM achieves lower values for the relative residual. Figure 11 reports the behavior of GSM and BGM applied to this problem perturbated with absolute noise. Again Figure 11(a) reports the relative residual of the smooth system ($\alpha = 0$). For small ($\alpha = 0.0001$, Figure 11(b)) and medium ($\alpha = 0.01$, Figure 11(c)) value of the noise both methods reach the same value of relative residual with GSM using clearly less evaluations of F than BGM. With a larger noise ($\alpha = 1$, Figure 11(c)), as for the proportional case, BGM is stalled at a higher value than GSM.

We have performed the same analysis on other problems, and observed a similar behavior, that is a systematically better robustness of GSM compared to the classic BGM when solving a noisy system of equations.

In summary, our method is more robust than BGM in the sense that it can solve noisy problems that BGM cannot. When both fail, GSM exhibits better decreases, which may be advantageous in practice.

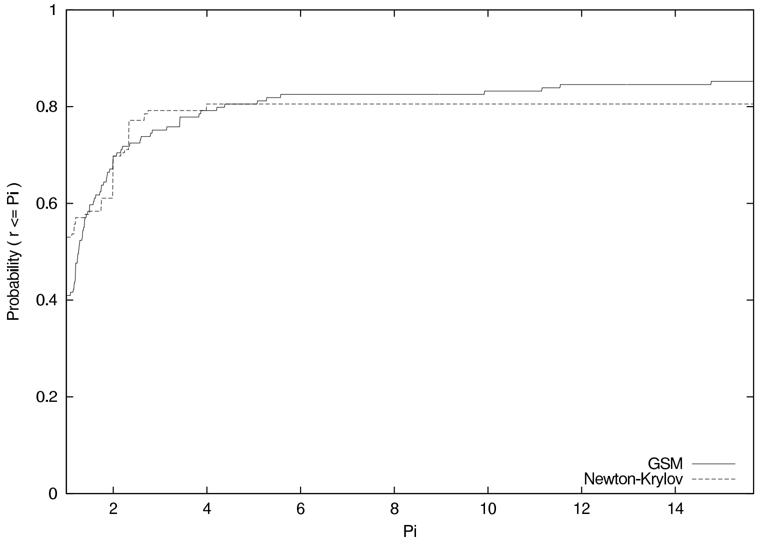


Figure 7: Performance profile – GSM and Newton-Krylov –

4.3 Large-scale problems

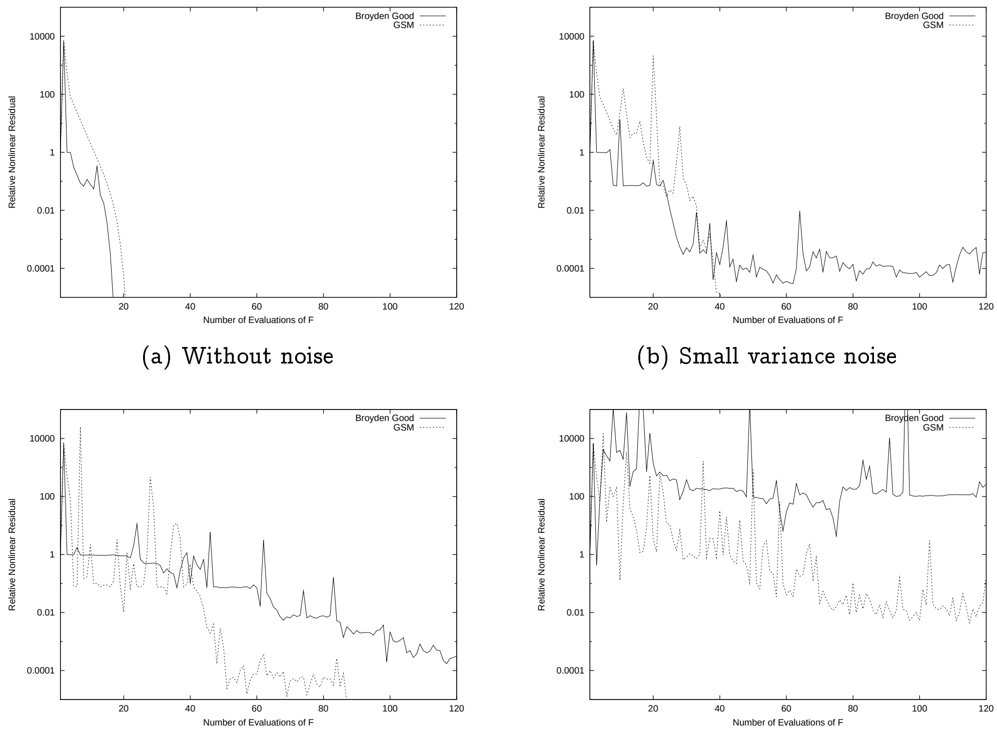
The main drawback of our approach is the relatively high cost in numerical linear algebra. Therefore it is particularly appropriate for medium-scale problems where F is very expensive to compute. Bierlaire and Crittin (forthcoming) propose an instance of this class of methods, designed to solve very large-scale systems of nonlinear equations without any assumption about the structure of the problem. The numerical experiments on standard large-scale problems show similar results: the algorithm outperforms classical large-scale quasi-Newton methods in terms of efficiency and robustness, its numerical performances are similar to the Newton-Krylov methods, and it is robust in presence of noise.

The complexity (both in time and memory) is linear in the size of the problem. Therefore, we were able to solve very large instances of a problem given by Spedicato and Huang (1997). The algorithm has been able to converge on a problem of size 2'000'000 in four hours and 158 iterations.

We are strongly interested in globalizing the large-scale version of our method. However, it requires future research to adapt our linesearch framework and to get an efficient globalization strategy in term of computational time.

5 Conclusion and perspectives

We have proposed a new class of generalized secant methods, based on the use of more than two iterates to identify the secant model. Contrarily to previous attempts for multi-iterate secant methods, the key ideas of this paper are (i) to use a least squares approach instead of an interpolation method to derive



(a) Without noise

(b) Small variance noise

(c) Medium variance noise

(d) Large variance noise

Figure 8: Behavior with proportional stochasticity

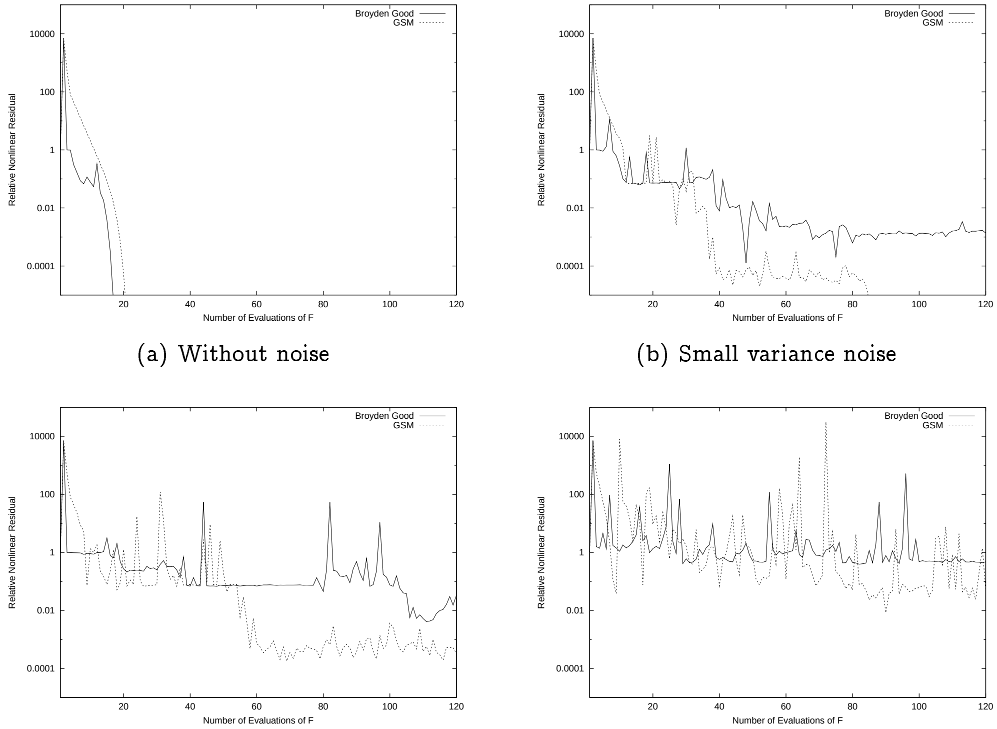
the secant model, and (ii) to explicitly control the numerical stability of the method.

A specific sub-class of this family of methods provides an update formula. We have proven the local convergence of an undamped quasi-Newton method based on this update formula. Moreover, we have performed extensive numerical experiments with several algorithms. The results show that our method produces significant improvement in term of robustness and number of function evaluations compared to classical methods. We have also shown that the globalization strategy presented in this paper significantly improves the robustness of quasi-Newton methods. Eventually, we have provided preliminary evidences that our method is more robust in the presence of noise in the function.

A theoretical analysis of a globally convergent version of our method must also be performed. We also conjecture that the local convergence rate is superlinear. And most importantly, the general behavior of the algorithm for solving noisy functions requires further analysis.

There are several variants of our methods that we plan to analyze in the future. Firstly, following Broyden's idea to derive BBM from (44), an update formula for B_{k+1}^{-1} can easily be derived in the context of our method:

$$B_{k+1}^{-1} = B_k^{-1} + \left(\Gamma\Gamma^T + Y_{k+1}\Omega^2 Y_{k+1}^T\right)^{-1} Y_{k+1}^T \Omega^2 \left(S_{k+1} - B_k^{-1} Y_{k+1}\right). \qquad (53)$$



(a) Without noise

(b) Small variance noise

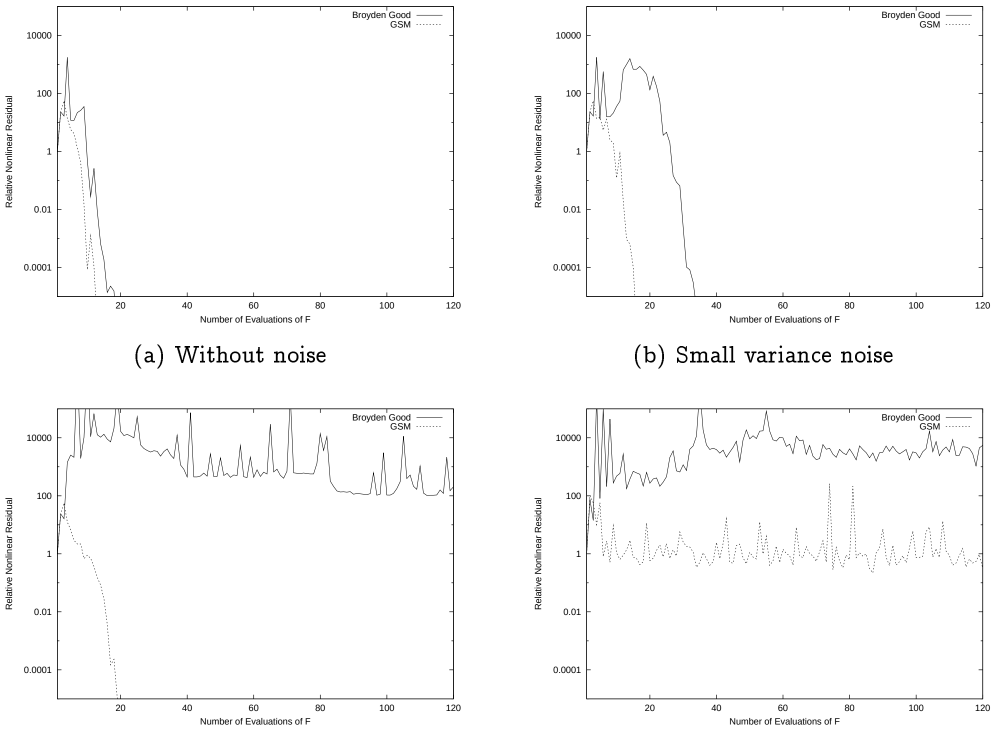
(c) Medium variance noise

(d) Large variance noise

Figure 9: Behavior with absolute stochasticity

From preliminary tests that we have performed, the “Good” and “Bad” versions of our method compare in a similar way as BGM and BBM. Secondly, non-update instances of our class of methods can be considered. In that case, the arbitrary matrix B_{k+1}^0 in (10) may be different from B_k. Choosing a matrix independent from k allows to use iterative scheme designed to solve large-scale least squares. In that case, choosing a matrix independent from k would allow to apply Kalman filtering (Kalman, 1960) to incrementally solve (10) and, consequently, improve the numerical efficiency of the method. For large scale problems, an iterative scheme such as LSQR (Paige and Saunders, 1982) can be considered. LSQR can also improve the efficiency of Kalman filter for the incremental algorithm (see Bierlaire and Crittin, 2004).

Finally, the ideas proposed in this paper can be tailored to optimization problems.



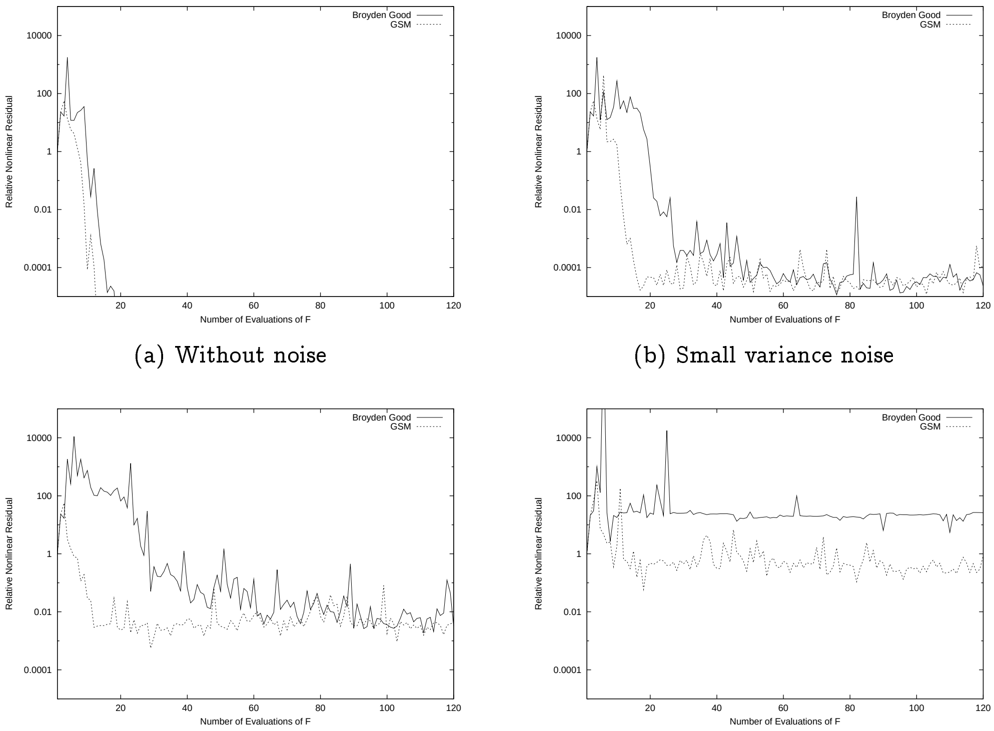
(a) Without noise

(b) Small variance noise

(c) Medium variance noise

(d) Large variance noise

Figure 10: Behavior with proportional stochasticity



(a) Without noise

(b) Small variance noise

(c) Medium variance noise

(d) Large variance noise

Figure 11: Behavior with absolute stochasticity

6 Appendix

6.1 Proof of Lemma 3

$$\begin{aligned}
\|S\Omega^2 S^T\| &\leq \|S\Omega\|^2 && (54)\\
&\leq \sum_{i=0}^{k} \|\omega_i s_i\|^2 && (55)\\
&\leq (k+1) \max_{i \in I_k} (|\omega_i| \|s_i\|)^2 && (56)\\
&\leq (k+1) \max_{i \in I_k} (|\omega_i| \|x_{k+1} - \tilde{x} + \tilde{x} - x_i\|)^2 && (57)\\
&\leq 2(k+1) \max_{i \in I_k} |\omega_i|^2 \max_{i \in I_{k+1}} \|x_i - \tilde{x}\|^2 && (58)\\
&\leq 2M_{\mathcal{P}} M_\omega^2 \max_{i \in I_{k+1}} \|x_i - \tilde{x}\|^2 && (59)
\end{aligned}$$

for all $\tilde{x} \in \mathbb{R}^{n \times n}$, in particular with $\tilde{x} = x^*$ which proves (21).

$$\begin{aligned}
\|\Omega^2 S^T\|^2 &\leq \sum_{i=0}^{k} \|\omega_i^2 s_i\|^2 && (60)\\
&\leq (k+1) \max_{i \in I_k} \Big(|\omega_i|^2 \|s_i\| \Big)^2 && (61)\\
&\leq (k+1) \max_{i \in I_k} |\omega_i|^4 \max_{i \in I_k} \|x_{k+1} - \tilde{x} + \tilde{x} - x_i\|^2 && (62)\\
&\leq 2(k+1) \max_{i \in I_k} |\omega_i|^4 \max_{i \in I_{k+1}} \|x_i - \tilde{x}\|^2 && (63)
\end{aligned}$$

for all $\tilde{x} \in \mathbb{R}^{n \times n}$. We obtain (22) with $\tilde{x} = x^*$:

$$\|\Omega^2 S^T\| \leq \sqrt{2M_{\mathcal{P}}} M_\omega^2 \max_{i \in I_{k+1}} \|x_i - x^*\| \qquad (64)$$

6.2 Proof of Lemma 4

Writing explicitly a column of the matrix $A = Y - J(x^*)S$

$$a_{\cdot j} = F(x_{k+1}) - F(x_i) - J(x^*)(x_{k+1} - x_{k-j+1}) \qquad (65)$$

with $a_{\cdot j}$ defining the column j of $A = (a_{ij})$.

Using (65) and Lemma 1 we can write:

$$\begin{aligned}
&\|Y - J(x^*)S\|^2\\
&\quad \leq \quad \sum_{j=1}^{k+1} \|a_{\cdot j}\|^2\\
&\quad \leq \quad (k+1) \max_{i \in I_k} \|F(x_{k+1}) - F(x_i) - J(x^*)(x_{k+1} - x_i)\|^2\\
&\quad \leq \quad (k+1) K_{lip}^2 \max_{i \in I_k} \left(\tfrac{\|x_i - x^*\| - \|x_{k+1} - x^*\|}{2} \|x_{k+1} - x_i\| \right)^2\\
&\quad \leq \quad 2(k+1) K_{lip}^2 \max_{i \in I_{k+1}} \|x_i - x^*\|^2 \max_{i \in I_{k+1}} \|x_i - x^*\|^2
\end{aligned}$$

Taking the square root on both sides:

$$\|Y - J(x^*)S\| \leq \sqrt{2M_{\mathcal{P}}} K_{lip} \max_{i \in I_{k+1}} \|x_i - x^*\|^2 \tag{66}$$

6.3 Proof of the Lemma 5

Let $A \in \mathbb{R}^{n \times n}$, we denote by $\lambda_m(A)$ and $\lambda_M(A)$ its smallest and largest eigenvalues, respectively. So we can write using the definition of the l_2 norm:

$$\begin{aligned} \|(\Gamma\Gamma^T + S\Omega^2 S^T)^{-1}\|_2 &= \lambda_M((\Gamma\Gamma^T + S\Omega^2 S^T)^{-1}) && (67) \\ &= \frac{1}{\lambda_m(\Gamma\Gamma^T + S\Omega^2 S^T)}. && (68) \end{aligned}$$

From assumption (A3), Γ^2 is computed using the modified Cholesky factorization, proposed by Schnabel and Eskow (1991), with parameter τ. Therefore,

$$\lambda_m(\Gamma\Gamma^T + S\Omega^2 S^T) \geq \tau, \tag{69}$$

which concludes the proof.

6.4 Description of the problem analyzed in Figures 8 and 9

The considered problem is the following system of equations:

$$f_i = x_i - \frac{\sum_{j=1}^4 x_j^3 + 1}{8} \qquad i = 1, \dots, 4 \tag{70}$$

with initial point $x_0 = (1.5, \dots, 1.5)$. The solution of this system is $x^* = (0.20432, \dots, 0.20432)$.

6.5 Description of the problem analyzed in Figures 10 and 11

The considered problem is the following system of equations of dimension n, where n is a positive multiple of 2.
For $i = 1, \dots, n/2$

$$\left\{ \begin{array}{rcl} f_{2i-1} &=& 10(x_{2i} - x_{2i-1}^2) \\ f_{2i} &=& 1 - x_{2i-1} \end{array} \right. \tag{71}$$

with initial point $x_0 = (-1.2, 1, \dots, -1.2, 1)$. The solution of this system is $x^* = (1, \dots, 1)$.

6.6 Linear problems in the tests set

We have tested three linear problems of the form $Ax = b$. They have been designed to challenge the tested algorithms.

1. For the first, the matrix A is the Hilbert matrix, and vector b is composed of all ones.

2. The second problem is based on the matrix A such that $a_{ij} = j$ if $i + j = n + 1$, and $a_{ij} = 0$ otherwise. All entries of the right-hand side b are -10. Its structure is designed so that the identitiy matrix is a poor approximation.

3. The third problem is based on a Vandermond matrix $A(\nu)$ with $\nu = (-1, -2, \ldots, -n)$. All entries of the right-hand side b are -1.

The starting point for all those problems is $x = (1, \ldots, 1)^T$.

References

Bierlaire, M. and Crittin, F. (2004). An efficient algorithm for real-time estimation and prediction of dynamic OD tables, *Operations Research* **52**(1): 116–127.

Bierlaire, M. and Crittin, F. (forthcoming). Solving noisy large scale fixed point problems and systems of nonlinear equations, *Transportation Science* .

Broyden, C. G. (1965). A class of methods for solving nonlinear simultaneous equations, *Mathematics of Computation* **19**: 577–593.

Broyden, C. G., Dennis, 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**: 233–246.

Choi, T. D. and Kelley, C. (2000). Superlinear convergence and implicit filtering, *SIAM Journal of Optimization* **10**.

Dennis, J. E. and Schnabel, R. B. (1996). *Numerical methods for unconstrained optimization and nonlinear equations*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa.

Dolan, E. D. and Moré, J. J. (2002). Benchmarking optimization software with performance profiles, *Mathematical Programming, Serie A* **91**: 201–213.

Eaton, J. W. (1997). GNU Octave: a high level interactive language for numerical computations, `www.octave.org`.

Eyert, V. (1996). A comparative study on methods for convergence acceleration of iterative vector sequences, *Journal of computational physics* **124**: 271–285.

Ford, J. A. (1999). A survey of multi-step quasi-Newton methods, *Proceedings of the International Conference on Scientific Computations*, Beirut, Lebanon.

Ford, J. A. and Moghrabi, I. A. (1997). Alternating multi-step quasi-Newton methods for unconstrained optimization, *Journal of Computational and Applied Mathematics* **82**: 105–116.

Gould, N. I. M., Leyffer, S. and Toint, P. L. (2005). A multidimensional filter algorithm for nonlinear equations and nonlineat least-squares, *SIAM Journal on Optimization* **15**(1): 17–38.

Gragg, W. and Stewart, G. (1976). A stable variant of the secant method for solving nonlinear equations, *SIAM Journal on Numerical Analysis* **13**: 889–903.

Johnson, D. D. (1988). Modified Broyden's method for accelerating convergence in self-consistent calculations, *Physical Review B* **38**(18): 12807–12813.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems, *J. of Basic Eng., Trans. ASME, Series D* **82**(1): 33–45.

Kelley, C. T. (2003). *Solving nonlinear equations with Newton's method*, SIAM, Philadelphia, Pa.

Martinez, J. M. (1979). Three new algorithms based on the sequantial secant method, *BIT* **19**: 236–243.

Martinez, J. M. (1982). Sobre dois metodos de broyden, *Matematica Aplicada e Computational* **1**.

Martinez, J. M. (2000). Practical quasi-Newton methods for solving nonlinear systems, *Journal of Computational and Applied Mathematics* **124**: 97–122.

Moghrabi, I. (1993). *Multi-step quasi-Newton methods for optimization*, PhD thesis, University of Essex, United Kingdom.

Moré, J. J., Garbow, B. S. and Hillstrom, K. E. (1981). Testing unconstrained optimization software, *ACM Transactions on Mathematical Software* **7**(1): 17–41.

Nocedal, J. and Wright, S. J. (1999). *Numerical optimization*, Operations Research, Springer Verlag, New-York.

Ortega, J. M. and Rheinboldt, W. C. (1970). *Iterative solution of nonlinear equations in several variables*, Academic Press, New York.

Paige, C. C. and Saunders, M. A. (1982). LSQR: an algorithm for sparse linear equations and sparse least squares, *ACM Transactions on Mathematical Software* **8**: 43–71.

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

Schnabel, R. B. and Eskow, E. (1991). A new modified Cholesky factorization, *SIAM Journal on Scientific and Statistical Computing* **11**: 1136–1158.

Schnabel, R. B. and Eskow, E. (1999). A revised modified Cholesky factorization, *SIAM Journal on Optimization* **9**: 1135–1148.

Spedicato, E. and Huang, Z. (1997). Numerical experience with Newton-like methods for nonlinear algebraic systems, *Computing* **58**: 69–99.

Vanderbilt, D. and Louie, S. G. (1984). Total energies of diamond (111) surface reconstructions by a linear combination of atomic orbitals method, *Physical Review B* **30**(10): 6118–6130.

Wolfe, P. (1959). The secant method for solving nonlinear equations, *Communications ACM* **12**: 12–13.