NUMERICAL ALGEBRA,
CONTROL AND OPTIMIZATION
Volume **1**, Number **1**, March **2011**

doi:10.3934/naco.2011.1.15

pp. **15–34**

RECENT ADVANCES IN NUMERICAL METHODS FOR NONLINEAR EQUATIONS AND NONLINEAR LEAST SQUARES

Ya-xiang Yuan

State Key Laboratory of Scientific/Engineering Computing
Institute of Computational Mathematics and Scientific/Engineering Computing
Academy of Mathematics and Systems Science, Chinese Academy of Sciences
Zhong Guan Cun Donglu 55, Beijing, 100190, China

Abstract. Nonlinear equations and nonlinear least squares problems have many applications in physics, chemistry, engineering, biology, economics, finance and many other fields. In this paper, we will review some recent results on numerical methods for these two special problems, particularly on Levenberg-Marquardt type methods, quasi-Newton type methods, and trust region algorithms. Discussions on variable projection methods and subspace methods are also given. Some theoretical results about local convergence results of the Levenberg-Marquardt type methods without non-singularity assumption are presented. A few model algorithms based on line searches and trust regions are also given.

1. **Introduction.** Nonlinear equations and nonlinear least squares have various applications in physics, chemistry, biology, engineering, economics, finance, and many other fields. These two special problems of mathematical optimization have been studied extensively and there are already many methods for solving them, such as Newton's method, quasi-Newton method, Gauss-Newton method, Levenberg -Marquardt method and trust region method, for example see [41, 42, 39, 10, 24, 38, 29, 30]. In recent years, large scale nonlinear equations and large scale nonlinear least squares have been attracting more and more attention from researchers, for example see [56, 19, 20]. Most large scale problems have either sparse property or special structure, therefore special approaches, such as partial separability and structure-exploiting, should be and can be applied to such problems [54, 55, 33, 5]. We will also discuss techniques that are useful for large scale problems, such as subspace techniques, variable projection methods and sparse quasi-Newton methods.

Nonlinear equations can be stated as:

$$F_i(x) = 0, \quad i = 1, ..., m, \tag{1.1}$$

where $F_i(x)(i = 1, ..., m)$ are m differentiable functions in $\Re^n$, and where n and m are positive integers. A closely related problem is to minimize the sum of the

2000 *Mathematics Subject Classification.* Primary: 65K10; Secondary: 90C05.

Key words and phrases. Nonlinear equations, nonlinear least squares, Levenberg-Marquardt, quasi-Newton, trust region, variable projection, subspace, local error bound conditions, convergence.

The author is supported by Chinese NSF grants 10831006, 11021101 and CAS grant kjcx-yw-s7.

squares of all the elements of $F(x) = (F_1(x), F_2(x), ..., F_m(x))^T$, namely

$$\min_{x\in\Re^n} \|F(x)\|_2^2 = \sum_{i=1}^{m} F_i(x)^2 \,. \tag{1.2}$$

Of course, (1.2) is an unconstrained optimization problem, which can be solved by any general methods for unconstrained optimization [14, 38, 53]. But, due to its special form, numerical methods for (1.2) should be constructed specially by exploiting its special structure.

If the nonlinear equations (1.1) have solutions, problems (1.1) and (1.2) are equivalent. Let the Jacobian matrix of $F(x)$ be

$$J(x) = (\nabla F(x))^T = [\nabla F_1(x), ..., \nabla F_m(x)]^T \in \Re^{m\times n}. \tag{1.3}$$

The linearization of equations (1.1) at an iterate point x_k is

$$F(x_k) + J(x_k)d = 0\,, \tag{1.4}$$

where $d = x - x_k$. When $m = n$ and $J(x_k)$ is non-singular, the above linear equations gives the Newton-Raphson iteration:

$$x_{k+1} = x_k - J(x_k)^{-1}F(x_k)\,. \tag{1.5}$$

If $m \neq n$ or $J(x_k)$ is singular, (1.4) may have no solutions. Hence, it is natural for us to replace (1.4) by the linear least squares problem:

$$\min_{d\in\Re^n} ||F(x_k) + J(x_k)d||_2^2\,, \tag{1.6}$$

which can be viewed as a linearization of (1.2). The least norm solution of the above subproblem is the Gauss-Newton step

$$d_k^{GN} = -(J(x_k))^+F(x_k)\,. \tag{1.7}$$

where $(J(x_k))^+$ is the Moore-Penrose generalized inverse of $J(x_k)$. If $J(x_k)$ is full column rank, the Gauss-Newton step can be written as

$$d_k^{GN} = -(J(x_k)^TJ(x_k))^{-1}J(x_k)^TF(x_k)\,. \tag{1.8}$$

Almost all the numerical methods can be viewed as some kinds of modification to the Newton-Raphson step or the Gauss-Newton step. This paper reviews recent advances on some methods for nonlinear equations and nonlinear least squares. This short review is by no means a complete survey. For example, we will not discuss interior point methods, which is also a very important class of methods for nonlinear equations and nonlinear least squares problems.

The paper is organized as follows. We discuss quasi-Newton methods and SQP methods in the next section, including recent results on sparse updates. Levenberg-Marquardt type methods and trust region algorithms are reviewed in Section 3, where we present recent results on the local convergence rates of Levenberg-Marquardt type methods under the local error bound conditions. In section 4 we discuss variable projection methods and subspace techniques are also briefly presented in Section 5.

2. **Quasi-Newton and SQP methods.** Consider the nonlinear equations problem (1.1), if we do not calculate the Jacobian matrix $J(x_k)$, and want to have an approximation to the Newton-Raphson step. The obvious approach is to use a quasi-Newton matrix B_k to replace the Jacobian. Thus, at each iteration a quasi-Newton direction d_k is computed by solving the following linearized system:

$$F(x_k) + B_k d = 0, \tag{2.1}$$

where $B_k \in \Re^{m\times n}$ is an approximation to the Jacobian matrix $J(x_k)$. Normally we require the following quasi-Newton condition

$$B_{k+1}s_k = y_k\,, \tag{2.2}$$

where

$$s_k = \alpha_k d_k \tag{2.3}$$
$$y_k = F(x_{k+1}) - F(x_k)\,, \tag{2.4}$$

and $\alpha_k > 0$ is computed by some line search techniques. Relation (2.2) is a secant condition. The quasi-Newton matrices B_k can be updated by different quasi-Newton update formulae. If we require that the increment $B_{k+1} - B_k$ is a rank 1 matrix, the general formula for B_{k+1} can be written as

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)u_k^T}{s_k^T u_k}\,, \tag{2.5}$$

for any vector $u_k \in \Re^n$ as long as $u_k^T s_k \neq 0$. The special choice of $u_k = s_k$ gives Broyden's unsymmetric Rank-1 formula:

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)s_k^T}{\|s_k\|_2^2}\,. \tag{2.6}$$

When $m = n$, we can also let $u_k = y_k - B_k s_k$, which gives the Broyden's symmetric rank-1 update:

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{s_k^T(y_k - B_k s_k)}\,. \tag{2.7}$$

There are many other updates, such as the Powell's symmetric Broyden, BFGS and DFP updates [14, 38, 53].

Replacing y_k by $J(x_{k+1})s_k$ in (2.5) leads to the following formula:

$$B_{k+1} = B_k + \frac{(J(x_{k+1}) - B_k)s_k u_k^T}{s_k^T u_k} \tag{2.8}$$

Of course, we do not assume that we have the full Jacobian matrix $J(x_{k+1})$ as otherwise we can simple set $B_{k+1} = J(x_{k+1})$. The formula (2.8) is useful in the case when we can compute the vector $J(x_{k+1})s_k$ by automatic differentiation are used, for example, see [21]. Similarly, we can apply the formula (2.8) to the B_{k+1}^T. Namely if we impose the adjoint tangent condition [22, 47]:

$$w_k^T B_{k+1} = w_k^T J(x_{k+1})\,, \tag{2.9}$$

we have the following update formula

$$B_{k+1} = B_k + \frac{v_k w_k^T (J(x_{k+1}) - B_k)}{v_k^T w_k} \tag{2.10}$$

for any two vectors v_k and w_k in $\Re^m$. If $v_k = w_k$, update (2.10) reduces to the adjoint Broyden update:

$$B_{k+1} = B_k + \frac{v_k v_k^T (J(x_{k+1}) - B_k)}{v_k^T v_k}. \tag{2.11}$$

Particular choices of v_k are given by [47]:

$$v_k = [J(x_{k+1}) - B_k]s_k \tag{2.12}$$
$$v_k = [F(x_{k+1}) - F(x_k)] - B_k s_k \tag{2.13}$$
$$v_k = F(x_{k+1}). \tag{2.14}$$

When v_k is given by (2.12)-(2.14), the quasi-Newton update (2.11) is called as adjoint Broyden tangent update, adjoint Broyden secant update and adjoint Broyden residual update respectively. Substituting (2.12) into (2.10), we obtain the two sided rank-1 update formula (see, [22, 47]):

$$B_{k+1} = B_k + \frac{(J(x_{k+1}) - B_k)s_k w_k^T (J(x_{k+1}) - B_k)}{w_k^T (J(x_{k+1}) - B_k)s_k}, \tag{2.15}$$

which can also be derived from (2.5) by setting $u_k = (J(x_{k+1}) - B_k)^T w_k$.

One good property of quasi-Newton updates for nonlinear equations is that they can also preserve the sparse structure of the original Jacobian matrices. Pioneer researches on sparse quasi-Newton methods were carried out by Schubert [50] and Toint [54, 55, 56].

Let the Jacobian matrix $J(x)$ has the following sparse property

$$(J(x))_{i,j} = 0, \qquad \text{if} (i,j) \in \mathcal{I} \tag{2.16}$$

where $\mathcal{I}$ is a subset of $\{\,(i,j)\,|\,i = 1, ..., m,\ j = 1, ..., n.\}$. For $i = 1, 2, ..., m$, define the vectors $s_k^{(i)}$ as follows:

$$\left(s_k^{(i)}\right)_j = \begin{cases} 0, & \text{if } (i,j) \in \mathcal{I}; \\ (s_k)_j, & \text{otherwise.} \end{cases} \tag{2.17}$$

Schubert's sparse Broyden Rank-1 update is

$$B_{k+1} = B_k + \sum_{i=1}^{m} e_i e_i^T \frac{(y_k - B_k s_k)(s_k^{(i)})^T}{s_k^T s_k^{(i)}}, \tag{2.18}$$

where e_i is the $i-$th unit vector in $\Re^m$, namely $(e_i)_i = 1$ and $(e_i)_j = 0$ for $j \neq i$. It is easy to verify that the update formula (2.18) satisfies the quasi-Newton condition (2.2) and has the sparsity property $(B_{k+1})_{i,j} = 0$ for $(i,j) \in \mathcal{I}$ provided that B_k has this sparsity property as well.

Schubert's sparse formula can be expressed more clearly by using orthogonal projections. Let the sparse structure of each row of the Jacobian matrix $J(x)$ be presented as a subspace in $\Re^n$:

$$V_i = \{x \in \Re^n \,|\quad x_j = 0, \text{ if } (i,j) \in \mathcal{I}\}. \tag{2.19}$$

And we denote the projection for $\Re^n$ to V_i by P_{V_i}, namely P_{V_i} is a diagonal matrix in $\Re^{n\times n}$ whose diagonal elements are given by

$$(P_{V_i})_{j,j} = \begin{cases} 0, & \text{if } (i,j) \in \mathcal{I}; \\ 1, & \text{otherwise.} \end{cases} \tag{2.20}$$

With these notations, Schubert's sparse Broyden update can be expressed as

$$B_{k+1} = B_k + \sum_{i=1}^{m} e_i e_i^T \frac{(y_k - B_k s_k) s_k^T P_{V_i}}{s_k^T P_{V_i} s_k}, \tag{2.21}$$

which can be viewed as the modification of the following row-by-row version of the Broyden formula:

$$B_{k+1} = B_k + \sum_{i=1}^{m} e_i e_i^T \frac{(y_k - B_k s_k) s_k^T}{s_k^T s_k}. \tag{2.22}$$

In fact, similar to the extension from (2.6) to (2.5), the above update formula can also be extended to

$$B_{k+1} = B_k + \sum_{i=1}^{m} e_i e_i^T \frac{(y_k - B_k s_k) \left(u_k^{(i)}\right)^T}{s_k^T u_k^{(i)}}, \tag{2.23}$$

where $u_k^{(i)} (i = 1, 2, ..., m)$ are vectors in $\Re^n$. Now, it is easy to see that Schubert's update (2.21) is just a special case of (2.23) with $u_k^{(i)} = P_{V_i} s_k$. Similarly, if we impose the quasi-Newton condition on the transpose:

$$w_k^T B_{k+1} = z_k^T, \tag{2.24}$$

we can have the following generalized update formula

$$B_{k+1} = B_k + \sum_{j=1}^{n} v_k^{(j)} \frac{(z_k^T - w_k^T B_k)}{w_k^T v_k^{(j)}} e_j e_j^T, \tag{2.25}$$

where $v_k^{(j)} \in \Re^m (j = 1, 2, ..., n)$ are vectors such that $w_k^T v_k^{(j)} \neq 0$. From the updates (2.23) and (2.25), we can obtain two new general sparse update formulae: the general sparse row-wise Broyden update

$$B_{k+1} = B_k + \sum_{i=1}^{m} e_i e_i^T \frac{(y_k - B_k s_k) \left(u_k^{(i)}\right)^T P_{V_i}}{s_k^T P_{V_i} u_k^{(i)}}, \tag{2.26}$$

and the general sparse column-wise Broyden update

$$B_{k+1} = B_k + \sum_{j=1}^{n} P_{\bar{V}_j} v_k^{(j)} \frac{(z_k^T - w_k^T B_k)}{w_k^T P_{\bar{V}_j} v_k^{(j)}} e_j e_j^T \tag{2.27}$$

where $P_{\bar{V}_j}$ is a diagonal matrix in $\Re^{m \times m}$ whose diagonal entries are

$$\left(P_{\bar{V}_j}\right)_{i,i} = \begin{cases} 0, & \text{if } (i,j) \in \mathcal{I}; \\ 1, & \text{otherwise.} \end{cases}. \tag{2.28}$$

Special cases of the above two updates (2.26) and (2.27) include

$$B_{k+1} = B_k + \sum_{i=1}^{m} e_i e_i^T \frac{(J(x_{k+1}) - B_k) s_k w_k^T (J(x_{k+1}) - B_k) P_{V_i}}{w_k^T (J(x_{k+1}) - B_k) P_{V_i} s_k} \tag{2.29}$$

$$B_{k+1} = B_k + \sum_{j=1}^{n} P_{\bar{V}_j} \frac{(J(x_{k+1}) - B_k) s_k w_k^T (J(x_{k+1}) - B_k)}{w_k^T P_{\bar{V}_j} (J(x_{k+1}) - B_k) s_k} e_j e_j^T \tag{2.30}$$

where were derived by Dai and Cheng [7]. Numerical results on these updates are studied in Cheng [6].

The sparsity condition (2.16) can be easily extended to structure condition:

$$(J(x))_{i,j} = J_{i,j}, \qquad \text{if } (i,j) \in \mathcal{I}, \tag{2.31}$$

namely some elements of the Jacobian are constants, which is quite common in mixed linear and nonlinear systems and separable/structured problems. In this case, we can also use (2.21) to update B_k as long as the initial matrix B_1 satisfies $(B_1)_{i,j} = J_{i,j}$ for all $(i,j) \in \mathcal{I}$.

Sparse or structured quasi-Newton matrices can also be used for problems without sparsity structures. For example, we can still use (2.21) even the Jacobian matrix $J(x)$ does not have any sparsity property. In this case, the quasi-Newton matrix remains unchanged for those elements whose indices are in $\mathcal{I}$. Specifically, if we let $\mathcal{I}$ to be all the indices except one column, it gives a one column per iteration quasi-Newton formula, which was first studied by Martinez [32].

Instead of solving the linear system (2.1) exactly, we can also obtain a d_k which satisfies (2.1) inexactly. This leads to inexact quasi-Newton methods for nonlinear equations, which use search direction d_k that satisfy

$$||B_k d + F(x_k)|| \leq \theta ||F(x_k)|| \tag{2.32}$$

where $\theta \in [0,1)$ is a given constant. For example, see [3, 4].

In optimization, the gradient method corresponds to using a scale quasi-Newton matrix. Due to the successful Barzilai and Borwein stepsizes for the gradient methods [1], scale quasi-Newton matrices have also been applied to nonlinear equations or nonlinear least squares [27, 28]. Consider the case $m = n$. We can let $B_k = \lambda_k I$ in (2.1), and use the week quasi-Newton condition

$$(x_{k+1} - x_k)^T B_k (x_{k+1} - x_k) = (x_{k+1} - x_k)^T [F(x_{k+1}) - F(x_k)] \tag{2.33}$$

to obtain the parameter λ_k.

Quasi-Newton methods can also be given by trying to approximate the inverse of the Jacobian instead of the Jacobian itself. Such quasi-Newton methods define the search direction d_k by

$$d_k = -H_k F(x_k)\,, \tag{2.34}$$

where H_k is an approximation to $J(x_k)^+$, and satisfies the following quasi-Newton condition

$$H_{k+1} y_k = s_k\,. \tag{2.35}$$

When $m = n$ and B_k is nonsingular, (2.1)-(2.2) is equivalent to (2.34)-(2.35).

Due to the classic Dennis and More theorem [9], quasi-Newton methods for nonlinear equations will convergence locally q-superlinearly if $J(x^*)$ is non-singular and

$$\lim_{k\to\infty} \frac{||(B_k - J(x^*))s_k||}{||s_k||} = 0\,. \tag{2.36}$$

Now we discuss quasi-Newton methods for nonlinear least squares. Consider the quadratic approximation to each of the nonlinear function $F_i(x)$ at x_k :

$$F_i(x_k + d) \approx F_i(x_k) + d^T F_i(x_k) + \frac{1}{2} d^T B_{k,i} d, \tag{2.37}$$

where $B_{k,i}$ is an approximation to $\nabla^2 F_i(x_k)$. Thus, we get the following problem

$$\min_{d\in\Re^n} \sum_{i=1}^{m} \left[F_i(x_k) + d^T F_i(x_k) + \frac{1}{2} d^T B_{k,i} d \right]^2, \tag{2.38}$$

which, if we ignore the higher order terms, leads to

$$\min_{d\in\Re^n} ||F(x_k)+J(x_k)d||_2^2+\frac{1}{2}d^TB_kd, \tag{2.39}$$

where B_k should be an approximation to $W(x_k)$ and $W(x)$ is defined by

$$W(x)=\sum_{i=1}^m F_i(x)\nabla^2F_i(x)\,. \tag{2.40}$$

Thus, a general SQP for nonlinear least squares uses the following search direction

$$d_k=-(J(x_k)^TJ(x_k)+B_k)^{-1}J(x_k)^TF(x_k)\,. \tag{2.41}$$

A good property of the above SQP step is that it is always a descent direction of $||F(x)||_2^2$ if B_k is positive definite, unless x_k is a stationary point of (1.2).

Another way to derive the SQP method is by first converting the nonlinear least squares problem into a standard constraint optimization problem and then applying a general purpose SQP method nonlinear optimization [48, 49]. The nonlinear least squares problem (1.2) can be reformulated as

$$\min_{(x,y)\in\Re^{n+m}} \quad \frac{1}{2}y^Ty \tag{2.42}$$

$$\text{s. t.} \quad F(x)-y=0\,. \tag{2.43}$$

The QP subproblem for the above equality constrained optimization problem is

$$\min_{(d_x,d_y)\in\Re^{n+m}} \quad y_k^Td_y+\frac{1}{2}(d_x^T,d_y^T)\bar{B}_k\begin{pmatrix} d_x\\ d_y\end{pmatrix} \tag{2.44}$$

$$\text{s. t.} \quad F(x_k)-y_k+J(x_k)d_x-d_y=0\,, \tag{2.45}$$

where $\bar{B}_k\in\Re^{(n+m)\times(n+m)}$ should be an approximation to the Hessian of the Lagrange function of problem (2.42)-(2.43), which is

$$\begin{pmatrix} \sum_{i=1}^n y_i\nabla^2F_i(x) & 0\\ 0 & I\end{pmatrix}. \tag{2.46}$$

It can be seen that the QP subproblem (2.44)-(2.45) is equivalent to (2.39) if

$$\bar{B}_k=\begin{pmatrix} B_k & 0\\ 0 & I\end{pmatrix}. \tag{2.47}$$

Let x^* be a local minimizer of (1.2), and assume the second order sufficient condition holds at x^*, namely $W(x^*)+J(x^*)J(x^*)^T$ is positive definite. Then, the superlinearly convergence condition for SQP step d_k obtained from (2.39) is

$$\lim_{k\to\infty}\frac{\|(B_k-W(x^*))d_k\|}{||d_k||}=0\,. \tag{2.48}$$

B_k can be updated by the BFGS formula

$$B_{k+1}=B_k-\frac{B_ks_ks_k^TB_k}{s_k^TB_ks_k}+\frac{y_ky_k^T}{s_k^Ty_k} \tag{2.49}$$

with $s_k=x_{k+1}-x_k$ and

$$y_k=(J(x_{k+1})-J(x_k))^TF(x_{k+1}) \tag{2.50}$$

or

$$y_k=J(x_{k+1}))^TF(x_{k+1})-J(x_k)^TF(x_k)-J(x_{k+1})^TJ(x_{k+1})s_k\,. \tag{2.51}$$

But neither of these two choices of y_k can always ensure $s_k^T y_k > 0$, a condition required by the BFGS update formula to maintain the positive definiteness of the quasi-Newton matrices. Hence Powell's technique [43] should be applied, namely y_k should be replaced by the following vector:

$$\bar{y}_k = \begin{cases} y_k, & \text{if } s_k^T y_k \geq 0.2 s_k^T B_k s_k; \\ \theta_k y_k + (1-\theta_k) B_k s_k, & \text{otherwise,} \end{cases} \tag{2.52}$$

where $\theta_k = 0.8 s_k^T B_k s_k / [s_k^T B_k s_k - s_k^T y_k]$.

Gauss-Newton method also be viewed as a special quasi-Newton method with $B_k = 0$ for all k. The special choice of $B_k = 0$ indicate that the computation cost of Gauss-Newton method per iteration is less than that of a general quasi-Newton method. For zero-residual problems, as $W(x)$ is a zero matrix at the solution, it seems that Gauss-Newton method will be as efficient as a quasi-Newton method. However, for nonzero residual problems, Gauss-Newton method may converge much slower than a quasi-Newton method. Hence, it is helpful to use some hybrid techniques [15]

The SQP method can be easily extended to nonlinear least squares problems with nonlinear constraints:

$$\min_{x \in \Re^n} \quad ||F(x)||_2^2 \tag{2.53}$$

$$\text{s. t.} \quad c_i(x) = 0, \quad i = 1, 2, ..., m_e; \tag{2.54}$$

$$c_i(x) \geq 0, \quad i = m_e + 1, ..., m_e + m_i \tag{2.55}$$

where m_e and m_i are the numbers of equality constraints and inequality constraints respectively. The QP subproblem for this constrained nonlinear least squares problem can be as follows.

$$\min_{d \in \Re^n} \quad ||F(x_k) + J(x_k)d||_2^2 + \frac{1}{2} d^T B_k d \tag{2.56}$$

$$\text{s. t.} \quad c_i(x_k) + d^T \nabla c_i(x_k) = 0, \quad i = 1, 2, ..., m_e; \tag{2.57}$$

$$c_i(x_k) + d^T \nabla c_i(x_k) \geq 0, \quad i = m_e + 1, ..., m_e + m_i \,. \tag{2.58}$$

Here, B_k should approximate to

$$\sum_{i=1}^{n} F_i(x) \nabla^2 F_i(x) - \sum_{i=1}^{m_e+m_i} \sigma_i \nabla^2 c_i(x) \,, \tag{2.59}$$

where σ_i $(i = 1, ..., m_e + m_i)$ are the estimations to the lagrange multipliers of (2.53)-(2.55). For example, it can be obtained by solving the following constrained linear least squares problem:

$$\min \quad \left\| J(x_k)^T F(x_k) - \sum_{i=1}^{m_e+m_i} \sigma_i \nabla c_i(x_k) \right\|_2^2 \tag{2.60}$$

$$\text{s. t.} \quad \sigma_i \geq 0, \quad i = m_e + 1, ..., m_e + m_i \,. \tag{2.61}$$

3. **Levenberg-Marquardt and trust region methods.** Levenberg-Marquardt method is a modification of the Gauss-Newton method, in which the search direction is defined by

$$d_k^{LM} = -(J(x_k)^T J(x_k) + \mu_k I)^{-1} J(x_k)^T F(x_k) \,, \tag{3.1}$$

where $\mu_k > 0$ is a parameter. For any fixed x_k, it can be easily shown that

$$\lim_{\mu_k \to 0+} d_k^{LM} = d_k^{GN} .$$

Classically, the parameter μ_k is updated based on the performance of the iterations. The general rule is that μ_k should be increased if the L-M step is not a good step, and reduced otherwise.

Nowadays it is more often to choose μ_k depending on $F(x_k)$ directly. For example,

$$\mu_k = \alpha ||F(x_k)||_2^\delta$$

Assumption 3.1. (a) The solution set of (1.1) is nonempty and denote it by X^*.

(b) $F(x)$ is continuously differentiable, and the Jacobi $J(x)$ is Lipschitz continuous on some neighbourhood of $x^* \in X^*$, i. e., there exist positive constants L_1 and $b_1 < 1$ such that

$$\|J(y) - J(x)\| \le L_1 \|y - x\|, \qquad \forall x, y \in N(x^*, b_1) = \{x \mid \|x - x^*\| \le b_1\} . \quad (3.2)$$

(c) $\|F(x)\|$ provides a local error bound on $N(x^*, b_1)$ for the system (1.1), i.e., there exists a constant $c_1 > 0$ such that

$$\|F(x)\| \ge c_1 \mathrm{dist}(x, X^*), \qquad \forall x \in N(x^*, b_1) . \quad (3.3)$$

Yamashita and Fukushima [58] established the following local convergence result.

Theorem 3.1. *Under the conditions of Assumption 3.1, if x_1 is chosen sufficiently close to x^* and $\mu_k = ||F(x_k)||^2$, then $x_{k+1} = x_k + d_k^{LM}$ converges to some the solution set X^* quadratically in the sense that*

$$dist(x_{k+1}, X^*) = O((dist(x_k, X^*)^2). \quad (3.4)$$

Fan and Yuan [11] improved the above result by allowing a wider range of the Levenberg-Marquardt parameter and strengthening the convergence to a fixed point.

Theorem 3.2. *Under the conditions of Assumption 3.1, if x_1 is chosen sufficiently close to X^* and $\mu_k = ||F(x_k)||^\delta$ with $\delta \in [1, 2]$, then $x_{k+1} = x_k + d_k^{LM}$ converges to some solution $\bar{x}$ of (1.1) quadratically, namely there exists $\bar{x} \in X^*$ such that*

$$\|x_{k+1} - \bar{x}\| = O(\|x_k - \bar{x}\|^2). \quad (3.5)$$

Globalization of the Levenberg-Marquardt method can be made by either line searches or trust regions. The following is a line search version of the Levenberg-Marquardt method.

Algorithm 3.1. (A Levenberg-Marquardt method with line search)

Step 1 Given $x_0 \in R^n, \delta \in [1, 2], \eta \in (0, 1), \varepsilon > 0, k := 0$.

Step 2 If $\|J_k^T F_k\| \le \varepsilon$ then stop; Set $\mu_k := \|F_k\|^\delta$; Compute d_k by (3.1).

Step 3 If d_k satisfies

$$\|F(x_k + d_k)\| \le \eta \|F(x_k)\|,$$

then $x_{k+1} = x_k + d_k$ otherwise $x_{k+1} = x_k + \alpha_k d_k$ where α_k is obtained by Wolfe or Armijo line search.

Step 4 $k := k + 1$; go to Step 2.

Theorem 3.3. *Assume that $F(x)$ is continuously differentiable. Let the sequence $\{x_k\}$ be generated by Algorithm 3.1. Then any accumulation point of $\{x_k\}$ is a stationary point of (1.2). Moreover, if an accumulation point x^* is a solution of nonlinear equation (1.1) and if Assumption 3.1 holds, then $\{x_k\}$ converges to x^* quadratically.*

The Levenberg-Marquardt method can also be regarded as a special SQP method with $B_k = \lambda_k I$, and it can be viewed as a penalty technique as it is essential adding a penalty term to restrict the stepsize. This technique can also be applied to nonlinear equations. Consider the special case when $n = m$, Newton-Raphson method (1.5) may not be well-defined as the Jacobian matrix $J(x_k)$ can be singular even for monotone problems. To overcome the difficulty caused by the possible singularity of $J(x_k)$, the regularized Newton method, which can also be called the Levenberg-Marquardt-Newton method, solves the following linear equations

$$(J(x_k) + \lambda_k I)d = -F(x_k) \tag{3.6}$$

to obtain the trial step d_k, where the regularization parameter $\lambda_k > 0$ is updated from iteration to iteration. (3.6) is well-defined for monotone problems, namely problems where $J(x)$ is positive semi-definite for all x. If the Jacobian is Lipschitz continuous and nonsingular at the solution and if the initial iterative point is chosen sufficiently close to the solution, then the trial steps satisfy

$$\|d_{k+1}\| \leq \kappa(\|d_k\|^2 + \lambda_k\|d_k\|) \tag{3.7}$$

for some positive κ [25]. Thus, if $\lambda_k = O(||d_k||)$, the local convergence rate of the regularized Newton step step (3.6) is quadratic.

However, the Levenberg-Marquardt type step d_k computed by (3.6) does not satisfy the linearized equation:

$$J(x_k)d + F(x_k) = 0 \tag{3.8}$$

which should be satisfied by $d = x^* - x_k$ when $F(x)$ is a linear function. The residual of linearized equation (3.8) is $-\lambda_k d_k$. Thus it is reasonable to solve the correction step $\hat{d}$ by

$$(J(x_k) + \lambda_k I)\hat{d} = \lambda_k d_k\,. \tag{3.9}$$

Let the correction step be denoted by $\hat{d}_k$, and define

$$s_k = d_k + \hat{d}_k\,. \tag{3.10}$$

Fan and Yuan [12] shows the following result:

Theorem 3.4. *Suppose that the conditions in Assumptions 3.1 hold and $\lambda_k = ||F(x_k)||$. If x_1 is chosen sufficiently close to the solution set X^* and $x_{k+1} = x_k + s_k$, then $\{x_k\}$ converges to some solution of (1.1) quadratically.*

The numerical results given by Fan and Yuan [12] indicate that the corrected step will improve the regularized Newton step. But we were not able to show theoretically the correct step will give a faster convergence rate, as one might guess whether it can lead to cubic convergence, improving the quadratic convergence of the regularized Newton step. Unfortunately this is not true, a simple example of $m = n = 1$ with $f(x) = x + 2x^2$ can be used to show that the Q-order of convergence of the regularized Newton method with correction step is only 2. A

cubic convergence can be achieved by an extra function evaluation at the $x_k + d_k$. If we define $\tilde{d}_k$ by solving

$$(J_k + \lambda_k I)\tilde{d} = -F(x_k + d_k)\,, \tag{3.11}$$

where d_k is the regularized Newton step by (3.6), the iterates generated by $x_{k+1} = x_k + d_k + \tilde{d}_k$ will converge cubically.

Theorem 3.5. *Suppose that the conditions in Assumptions 3.1 hold and* $\lambda_k = ||F(x_k)||$. *If* x_1 *is chosen sufficiently close to the solution set* X^* *and* $x_{k+1} = x_k + d_k + \tilde{d}_k$, *then* $\{x_k\}$ *converges to some solution of (1.1) cubically.*

Proof. (3.6) gives that

$$F(x_k+d_k) = F(x_k)+J(x_k)d_k+O(||d_k||^2) = -d_k||F(x_k)||+O(||d_k||^2) = O(||d_k||^2)\,. \tag{3.12}$$

It follows from the above relation and $(J(x_k) + \lambda_k I)\tilde{d}_k = -F(x_k + d_k)$ that

$$||\tilde{d}_k|| = O(||d_k||^2)\,. \tag{3.13}$$

This shows that $\tilde{d}_k$ is a second order correction step. Moreover, we have that

$$\begin{aligned} F(x_k + d_k + \tilde{d}_k) &= F(x_k + d_k) + J(x_k + d_k)\tilde{d}_k + O(||\tilde{d}_k||^2) \\ &= F(x_k + d_k) + J(x_k)\tilde{d}_k + O(||d_k||||\tilde{d}_k||) \\ &= -\lambda_k\tilde{d}_k + O(||d_k||||\tilde{d}_k||) = O(||d_k||||\tilde{d}_k||)\,. \end{aligned} \tag{3.14}$$

This shows that

$$||F(x_{k+1})|| = O(||d_k||^3)\,. \tag{3.15}$$

This relation and the fact that $||d_k|| = O(||F_k||)$ imply that $||d_{k+1}|| = O(||d_k||^3)$ which shows that the sequence x_k converges cubically. □

The Levenberg-Marquardt step (3.1) is the solution of the following subproblem [34, 35]

$$\min_{d\in\Re^n} \quad ||J(x_k)d + F(x_k)||_2^2 \tag{3.16}$$

$$\text{s. t.} \quad ||d||_2 \le ||d_k^{LM}||_2\,. \tag{3.17}$$

This constraint in the above subproblem has trust region flavor as it restricts the step-length. If we replace the linearized least squares objective function in (3.16) by the more accurate approximate model (2.39), we obtain the standard SQP trust region subproblem for nonlinear least squares ([40, 34]):

$$\min_{d\in\Re^n} \quad ||F(x_k) + J(x_k)d||_2^2 + \frac{1}{2}d^T B_k d, \tag{3.18}$$

$$\text{s. t.} \quad ||d||_2 \le \Delta_k\,, \tag{3.19}$$

where $\Delta_k > 0$ is the trust region bound, which is updated from iteration to iteration. Hence a SQP-trust region algorithm for nonlinear least squares (1.2) can be given as follows.

Algorithm 3.2. (SQP Trust Region Algorithm for NLS)

Step 1 Given $x_1 \in \Re^n$, Choose $\Delta_1 > 0$, $c_0 \in [0, 1)$, $c_1 > 1 > c_4 > c_3$ and $c_2 \in [c_0, 1)$. and $\epsilon > 0$ sufficiently small, k=1.

Step 2 If $||J_k^T F_k|| < \epsilon$, then stop.

Step 3 Compute s_k by (3.18)-(3.19);
Compute the ratio

$$r_k = \frac{||F(x_k)||_2^2 - ||F(x_k + s_k)||_2^2}{||F(x_k)||_2^2 - ||F(x_k) + J(x_k)s_k||_2^2 - \frac{1}{2}s_k^T B_k s_k}\,. \tag{3.20}$$

Step 4

$$x_{k+1} = \begin{cases} x_k + s_k & \text{if } r_k \geq c_0 \\ x_k & \text{otherwise,} \end{cases} \tag{3.21}$$

$$\Delta_{k+1} \in \begin{cases} [c_3||s_k||, \quad c_4\Delta_k] & \text{if } r_k < c_2 \\ [\Delta_k, \quad c_1\Delta_k] & \text{otherwise} \end{cases}, \tag{3.22}$$

Step 5 $k := k + 1$, goto Step 2.

The quadratic model function in (3.18) can be replaced by other quadratic approximations to $||F(x_k + d)||_2^2$. For example, one can use interpolation technique to obtain derivative-free trust region algorithms[63, 64]

The above algorithm can also used for solving nonlinear equations (1.1). Moreover, for (1.1), we can use other norms such as L_1 and L_∞ norms [13, 44, 8, 59]. For example, if we use the L_1 norm, we can have the following L_1 trust region subproblem

$$\min_{d\in\Re^n} \qquad ||F(x_k) + J(x_k)d||_1 + \frac{1}{2}d^T B_k d, \tag{3.23}$$

$$\text{s. t.} \qquad ||d||_\infty \leq \Delta_k\,, \tag{3.24}$$

which was given by Fletcher [13].

The Levenberg-Marquardt step (3.1) is the solution of the following subproblem:

$$\min_{d\in\Re^n} ||F(x_k) + J(x_k)d||_2^2 + \mu_k||d||_2^2\,. \tag{3.25}$$

Thus, an extension of the Levenberg-Marquardt method is to consider the subproblem of the following form:

$$\min_{d\in\Re^n} \sqrt{||F(x_k) + J(x_k)d||_2^2 + \mu_k||d||_2^2} + \sigma_k||d||_2^2\,. \tag{3.26}$$

More details can be found in [37, 2].

The Levenberg-Marquardt and SQP methods discussed in this section can be easily extended to problems with convex constraints, for example, see Kanzow, Yamashita and Fukushima [26].

4. **Variable projection methods.** Consider the special separable nonlinear least squares problem:

$$\min_{a\in R^p, b\in R^q} \psi(a, b) = \frac{1}{2}||\ y(b) - \Phi(b)a||_2^2, \tag{4.1}$$

where $y : R^q \mapsto R^m$; $\Phi : R^q \mapsto R^{m\times p}$, with $(\Phi(b))_{ij} = \phi_j(b, t_i)$, $i = 1, 2, ..., m$, $j = 1, 2, ...p$.

Golub and Pereyra [16] proposed the variable projection method for (4.1) when $y(b) = y_0$. The main idea of their variable projection method is eliminating the variable a to obtain a nonlinear least squares problem only depending on variable b, and applying standard nonlinear least squares methods such as Gauss-Newton method for the reduced problem.

For any fixed $b \in R^q$, (4.1) reduces to a linear least squares problem and it can be solved by choosing a to be the least-norm solution

$$\hat{a}(b) = \Phi^+(b)y(b)\,. \tag{4.2}$$

Substituting (4.2) into (4.1), we obtain

$$\min_{a\in R^p, b\in R^q} \psi(a,b) = \min_{b\in R^q} \frac{1}{2}||\ y(b) - \Phi(b)\Phi^+(b)y(b)||_2^2 = \min_{b\in R^q} \frac{1}{2}||P^{\perp}_{\Phi(b)}y(b)||_2^2\,. \tag{4.3}$$

Here, $P^{\perp}_{\Phi(b)}$ is the orthogonal projector from R^m to the null space of $\Phi(b)^T$. Thus, the Guass-Newton method for (4.3) is

$$b_{k+1} = b_k - [(\nabla(P^{\perp}_{\Phi(b)}y(b))^T]^+ P^{\perp}_{\Phi(b)}y(b)\,. \tag{4.4}$$

Golub and Pereyra [16] develops techniques to compute the Fréchet derivative of the orthogonal projector and its general inverse, so that the corresponding Gauss-Newton step can be computed. The Levenberg-Marquardt technique is also used in their algorithm. Thus, the iterate formula of the Golub and Pereyra method [16, 17] is

$$b_{k+1} = b_k - (P^{\perp}_{\Phi(b_k)}[y'(b_k) - \Phi'(b_k)\hat{a}(b_k)] + P_{\Phi(b_k)}\Phi^+(b_k)\Phi'(b_k)\varphi_k))^+\varphi_k\,, \tag{4.5}$$

where $\hat{a}(b)$ is defined by (4.2) and $\varphi_k = P^{\perp}_{\Phi(b_k)}y(b_k)$. Kaufman[23] simplified the Jacobian formula of the orthogonal projector, and obtains

$$b_{k+1} = b_k - (P^{\perp}_{\Phi(b_k)}[y'(b_k) - \Phi'(b_k)\hat{a}(b_k)])^+\varphi_k\,. \tag{4.6}$$

Later, Ruhe and Wedin [45] extended the variable projection idea to the general nonlinear case. They considered and developed the implicit reduction methods for the separable nonlinear least squares problem (4.1) by adopting alternate iteration between the two sets of the variables.

Liu and Yuan [31] proposed an inexact Newton method for the separable problem. Their approach is to update the two set of variables at the same time instead of alternately updating a and b as in the Ruhe and Wedin's approach. The motivation of their technique is to exploit the special structure of the separable problem in order to obtain a method which is as close as possible to the Newton's method without computing any second order derivatives. Another motivation of their approach is that it seems more efficient to update both sets of variables a and b together than to update them alternately.

Define

$$\varphi(a,b) = y(b) - \Phi(b)a. \tag{4.7}$$

Direct calculations show that

$$\varphi_a(a,b) = \Phi(b), \quad \varphi_b(a,b) = y'(b) - \Phi'(b)a, \tag{4.8}$$

and

$$\varphi_{ab}(a,b) = \Phi'(b)^T, \quad \varphi_{ba}(a,b) = \Phi'(b), \quad \varphi_{aa}(a,b) = 0. \tag{4.9}$$

Hence, without computing any second order derivatives of the original functions, the second order derivatives of $\varphi(a,b)$ is known except the term $\varphi_{bb}(a,b)$. This is very helpful for us to give an inexact Newton method for (4.1). The Newton's method for (4.1) is

$$\begin{pmatrix} a_{k+1} \\ b_{k+1} \end{pmatrix} = \begin{pmatrix} a_k \\ b_k \end{pmatrix} - [\nabla^2\psi(a_k,\, b_k)]^{-1}\nabla\psi(a_k,\, b_k)\,. \tag{4.10}$$

The gradient is easy to compute,

$$\nabla\psi(a_k,\, b_k) = \begin{pmatrix} \varphi_a^T\varphi \\ \varphi_b^T\varphi \end{pmatrix}. \tag{4.11}$$

For the second order derivatives, we have that, due to (4.9),

$$\nabla^2\psi = \begin{pmatrix} \varphi_a^T\varphi_a & \varphi_a^T\varphi_b + \Phi'(b)\varphi \\ \varphi_b^T\varphi_a + (\Phi'(b)\varphi)^T & \varphi_b^T\varphi_b + \varphi_{bb}^T\varphi \end{pmatrix}. \tag{4.12}$$

Now consider any iterate method having the following form:

$$x_{k+1} = x_k - B_k^{-1}\nabla\psi(a_k, b_k)\,, \tag{4.13}$$

where B_k is an $n \times n$ matrix. Because Newton's method converges quadratically, intuitively, a condition for an efficient method for (4.1) in the form of (4.13) should be that B_k is a good approximation to the Hessian matrix (4.12). Under the general unseparated scheme, the Gauss-Newton method approximates the Hessian by the first order term of φ as

$$B^{GN} = \begin{pmatrix} \varphi_a^T\varphi_a & \varphi_a^T\varphi_b \\ \varphi_b^T\varphi_a & \varphi_b^T\varphi_b \end{pmatrix}. \tag{4.14}$$

By comparing (4.14) and (4.12) we can easily find that there is a term $\Phi'(b)\varphi$, neglected by (4.14), which can be obtained without computing any second order derivatives. Therefore, it seems that it is more reasonable to use

$$B^{(0)} = \begin{pmatrix} \varphi_a^T\varphi_a & \varphi_a^T\varphi_b + \Phi'(b)\varphi \\ \varphi_b^T\varphi_a + (\Phi'(b)\varphi)^T & \varphi_b^T\varphi_b \end{pmatrix}, \tag{4.15}$$

to approximate the Hessian matrix (4.12). Unfortunately, it is not a good idea, because $B^{(0)}$ may not be positive semi-definite, even when the real Hessian $\nabla^2\psi$ is positive definite. Theoretical analysis given by Liu and Yuan [31] show that $B^{(0)}$ based scheme is not as good as Gauss-Newton method in most cases.

Now we try to modify (4.15) so that the positive definite condition is satisfied. First, we define $\varphi_C = (\varphi_a^+)^T\Phi'(b)\varphi$ and replace $\varphi_a^T\varphi_b + \Phi'(b)\varphi$ by $\varphi_a^T(\varphi_b + \varphi_C)$. Furthermore, if the lower right block term $\varphi_b^T\varphi_b$ is replaced by $(\varphi_b+\varphi_c)^T(\varphi_b+\varphi_c)$, we would have the following approximation formular:

$$B^{(1)} = \begin{pmatrix} \varphi_a^T\varphi_a & \varphi_a^T(\varphi_b + \varphi_c) \\ (\varphi_b + \varphi_c)^T\varphi_a & (\varphi_b + \varphi_C)^T(\varphi_b + \varphi_C) \end{pmatrix}. \tag{4.16}$$

Now, we can see $B^{(1)}$ is always positive semi-definite and has more second-order information than B^{GN}. If $\Phi(b)$ has rank p, we can show that

$$\varphi_a^T\varphi_b + \Phi'(b)\varphi = \varphi_a^T(\varphi_b + \varphi_C). \tag{4.17}$$

In this case, we have

$$B^{(1)} = \nabla^2\psi + \begin{pmatrix} 0 & 0 \\ 0 & \varphi_b^T\varphi_C + \varphi_C^T\varphi_b + \varphi_C^T\varphi_C - \varphi_{bb}^T\varphi \end{pmatrix}. \tag{4.18}$$

Hence the error matrix $B^{(1)} - \nabla^2\psi$ is a rank q matrix. Results in Liu and Yuan [31] indicate that (4.16) is a better approximation to $\nabla^2\psi$ than B^{GN}. Adding $\varphi_C^T\varphi_C$ to the bottom-right part of (4.16), we get

$$B^{(2)} = \begin{pmatrix} \varphi_a^T\varphi_a & \varphi_a^T(\varphi_b + \varphi_c) \\ (\varphi_b + \varphi_c)^T\varphi_a & (\varphi_b + \varphi_C)^T(\varphi_b + \varphi_C) + \varphi_C^T\varphi_C \end{pmatrix}, \tag{4.19}$$

which is another better approximation to $\nabla^2\psi$. Methods based on approximate Hessian (4.16) or (4.19) for separable nonlinear least squares problem (4.1) are given by Liu and Yuan [31], and one of them is listed as follows.

Algorithm 4.1. (Structured Unseparated Method: Trust Region Version)

Step 1 Given $x_1 = (a_1, b_1) \in \Re^n$, $\Delta_1 > 0$, $c_1 > 1$, $c_2 > 1$, $0 < p_1 < p_2 < 1$; and $\epsilon > 0$ sufficiently small, k=1.

Step 2 If $||\nabla\psi(x_k)|| < \epsilon$, then stop.

Step 3 Compute B_k by either (4.16) or (4.19), solve

$$\min_{d\in\Re^n} \quad m_k(d) = \nabla\psi(x_k)^T d + \frac{1}{2} d^T B_k d, \tag{4.20}$$

$$\text{s.t.} \quad ||d||_2 \le \Delta_k, \tag{4.21}$$

obtaining d_k.

Step 4 Compute the ratio

$$r_k = \frac{\psi(x_k) - \psi(x_k + d_k)}{m(0) - m(d_k)}, \tag{4.22}$$

Generate Δ_{k+1} and x_{k+1}:

$$\Delta_{k+1} = \begin{cases} \frac{1}{c_1}\|d_k\|_2 & \text{if } r_k \le p_1, \\ \max[c_2\|d_k\|_2,\ \Delta_k] & \text{if } r_k \ge p_2, \\ \Delta_k & \text{otherwise;} \end{cases} \tag{4.23}$$

$$x_{k+1} = \begin{cases} x_k + d_k & \text{if } r_k > 0, \\ x_k & \text{otherwise.} \end{cases} \tag{4.24}$$

Step 5 $k := k + 1$, goto Step 2.

In Step 3 of the above algorithm, d_k can be either an exact solution[36] or an approximation solution of subproblem (4.20)-(4.21) obtained by the truncated conjugate gradient method [55, 51]. The step d_k computed by the truncated conjugate gradient method has the nice property that it reduces the objective function by at least half of the reduction of the exact solution[60].

5. **Subspace methods.** Subspace technique has been used well in numerical linear algebra [18, 46]. It can also be used to construct numerical methods for optimization [52, 57, 61, 62]. In this section, we briefly discuss subspace techniques for nonlinear equations and nonlinear least squares problems.

First, we consider subspace techniques for nonlinear equations (1.1). At the $k-$th iteration, instead of solving (2.1), we let $d_k = Q_k z_k$ by solving z_k from

$$P_k^T [\ F(x_k) + J_k Q_k z\ ] = 0, \tag{5.1}$$

where

$$P_k = [\ p_1^{(k)},\ p_2^{(k)},\ ...,\ p_{i_k}^{(k)}\ ] \tag{5.2}$$

$$Q_k = [\ q_1^{(k)},\ q_2^{(k)},\ ...,\ q_{i_k}^{(k)}\ ] \tag{5.3}$$

are two full column rank matrices and where $i_k \in [1, n]$ is an integer. Normally, i_k is chosen as i_{k-1} or $i_{k-1} + 1$, and it is generally the case that i_k is much less than n, particularly for large scale problems.

The reduced linear system (5.1) defines a step d_k in the subspace

$$\mathcal{S}_k = Span\{q_1^{(k)}, q_2^{(k)}, ..., q_{i_k}^{(k)}\}, \tag{5.4}$$

with the property that the residual of the original linear system (1.4) orthogonal to the subspace spanned by $p_j (j = 1, ..., i_k)$.

Thus, we do not require the full Jacobian matrix J_k for obtaining a search direction in the subspace $\mathcal{S}_k$. Instead, we only need M_k to be an approximation to the reduced Jacobian $P_k^T J_k Q_k$, which is a square matrix in $\Re^{i_k \times i_k}$, and which normally has much fewer elements than the full Jacobian matrix J_k. A general subspace method for nonlinear equations can be described as follows.

Algorithm 5.1. (A general subspace algorithm for nonlinear equations)

Step 1 Given x_1, $\epsilon > 0$, $k := 1$.

Step 2 Generate P_k, Q_k and $M_k \approx P_k^T J(x_k) Q_k$;
Compute the vector z_k by solving:

$$P_k^T F(x_k) + M_k z = 0, \tag{5.5}$$

Set $d_k = Q_k z_k$. If $\|d_k\| \leq \epsilon$ then stop.

Step 3 Carry out a line search, obtaining a stepsize α_k and set

$$x_{k+1} = x_k + \alpha_k d_k. \tag{5.6}$$

Step 4 $k := k + 1$, Go to Step 2.

Of course, we can also use the quasi-Newton techniques discussed in Section 2 to generate M_k. If $\mathcal{S}_{k+1} = \mathcal{S}_k$, we assume that $Q_{k+1} = Q_k$ and $P_{k+1} = P_k$. Consequently, the corresponding quasi-Newton condition is

$$M_{k+1} z_k = P_k^T y_k \,. \tag{5.7}$$

For example, Broyden's unsymmetric rank-1 formula has the following subspace version:

$$M_{k+1} = M_k + \frac{(P_k^T y_k - M_k z_k) z_k^T}{||z_k||_2^2} \,. \tag{5.8}$$

In the case $i_{k+1} = i_k + 1$, it is normally that $\mathcal{S}_{k+1} = Span\{\mathcal{S}_k, \bar{q}_k\}$ for some vector $\bar{q}_k$. Let us assume that $q_j^{(k+1)} = q_j^{(k)}$ and $p_j^{(k+1)} = p_j^{(k)}$ for $j = 1, 2, ..., i_k$ and $q_{i_{k+1}}^{(k+1)}$ is orthogonal to $\mathcal{S}_k$. Thus, again taking Broyden's unsymmetric rank-1 for example, M_{k+1} can be defined by

$$M_{k+1} = \begin{pmatrix} \bar{M}_k & 0 \\ 0 & \mu_k \end{pmatrix}$$

with

$$\bar{M}_k = M_k + \frac{(P_k^T y_k - M_k z_k) z_k^T}{||z_k||_2^2} \,, \tag{5.9}$$

and where μ_k is an estimation to $(p_{i_{k+1}}^{(k+1)})^T J(x_{k+1}) q_{i_{k+1}}^{(k+1)}$.

A merit function have to be used for the line search in Step 3 of the above algorithm. In general, it should be some norm of the vector $F(x)$, such as $||F(x)||_2$, $||F(x)||_1$ and $||F(x)||_\infty$. For a given merit function, P_k and Q_k should be chosen accordingly so that the search direction obtained from (5.5) is a descent direction of the corresponding merit function.

Now, we consider subspace techniques for the nonlinear least square problems (1.2). It is natural to consider the following subspace subproblem:

$$\bar{Q}_k(z) = \|F(x_k) + J_k Q_k z\|_2^2 + z^T B_k z, \tag{5.10}$$

where $B_k \in \Re^{i_k \times i_k}$ approximates the reduced matrix

$$Q_k^T W_k Q_k = \sum_{i=1}^{m} F_i(x_k) Q_k^T \nabla^2 F_i(x_k) Q_k. \tag{5.11}$$

We can give a subspace algorithm for nonlinear least squares as follows.

Algorithm 5.2. (A subspace trust region algorithm for nonlinear least squares)

Step 1 Given $x_1 \in \Re^n$, $\Delta_1 > 0$, Choose matrices Q_1 and B_1 , Given $\epsilon > 0$, $k := 1$.

Step 2 Solve the subspace subproblem:

$$\min_{z \in \Re^{|i_k|}} \quad \bar{Q}_k(z) = \|F(x_k) + J_k Q_k z\|_2^2 + z^T B_k z \tag{5.12}$$

$$\text{s. t.} \quad \|z\|_2 \le \Delta_k, \tag{5.13}$$

obtaining z_k, set $s_k = Q_k z_k$. If $\|s_k\| \le \epsilon$ then stop.

Step 3 Define

$$r_k = \frac{\|F(x_k)\|_2^2 - \|F(x_k + s_k)\|_2^2}{\bar{Q}_k(0) - \bar{Q}_k(z_k)}. \tag{5.14}$$

Set

$$x_{k+1} = \begin{cases} x_k + s_k & \text{if } f(x_k + s_k) < f(x_k); \\ x_k & \text{otherwise.} \end{cases} \tag{5.15}$$

$$\Delta_{k+1} = \begin{cases} \frac{1}{2}\|z_k\|_2 & \text{if } r_k < 0.1; \\ 2\Delta_k & \text{if } r_k > 0.9 \text{ and } 2\|z_k\| > \Delta_k \ ; \\ \Delta_k & \text{otherwise.} \end{cases} \tag{5.16}$$

Step 4 Generate Q_{k+1} and B_{k+1}.

Step 5 $k := k + 1$, Go to Step 2.

Unless the gradient vector $J_k^T F_k$ is orthogonal to the subspace $\mathcal{S}_k$, the trust region subproblem (5.12)-(5.13) will also give a non-zero trial step $s_k = Q_k z_k$, and furthermore, s_k is always a descent direction of $\|F(x)\|_2^2$ at x_k.

Instead of imposing the trust region constraint to z_k, we can also directly require the trial step in the original space to satisfy the trust region condition. Namely we can modify the statements of Algorithm 4 slightly so that s_k is the solution of

$$\min_{s \in \mathcal{S}_k} \quad \hat{Q}_k(s) = \|F(x_k) + J_k s\|_2^2 + s^T \hat{B}_k s, \tag{5.17}$$

$$\text{s. t.} \quad \|s\|_2 \le \Delta_k. \tag{5.18}$$

Thus any orthogonal basis of $\mathcal{S}_k$ consists of a matrix Q_k. We can choose the Q_k properly so that the subproblem (5.17)-(5.18) is easy to solve and the approximate matrix $B_k = Q_k^T \hat{B}_k Q_k$ is easy to obtain.

For both nonlinear equations and nonlinear least squares problems, the essential part of a subspace method is the choice of the subspace $\mathcal{S}_k$. Subspace techniques for unconstrained optimization can be borrowed here. For example, possible choices are as follows:

$$\mathcal{S}_k = Span\{F_1, F_2, ..., F_{k-1}, F_k\} \tag{5.19}$$

$$\mathcal{S}_k = Span\{J_k^+ F(x_k), s_{k-1}, s_{k-2}, ..., s_2, s_1\} \tag{5.20}$$

$$\mathcal{S}_k = Span\{J_k^T F_k, J_{k-1}^T F_{k-1}, ..., J_2^T F_2, J_1^T F_1\}. \tag{5.21}$$

In order to prevent the dimension of the subspace getting larger and larger, we can also use

$$\mathcal{S}_k = Span\{F_{k-M}, F_2, ..., F_{k-1}, F_k\} \quad (5.22)$$

$$\mathcal{S}_k = Span\{J_k^+ F(x_k), s_{k-1}, s_{k-2}, ..., s_{k-M+1}, s_{k-M}\} \quad (5.23)$$

$$\mathcal{S}_k = Span\{J_k^T F_k, J_{k-1}^T F_{k-1}, ..., J_{k-M+1}^T F_{k-M+1}, J_{k-M}^T F_{k-M}\} \quad (5.24)$$

for some fixed positive integer M when $k > M$, Other choices of $\mathcal{S}_k$ are also possible, for example we can replace $-J_k^+ F(x_k)$ in (5.20) by a randomly generated unite vector or any descent coordinate direction.

6. **Acknowledgement.** This article was written during my visit at Bayreuth University supported by Alexander-von-Humboldt Foundation. I would like to thank Humboldt Foundation for the support and thank Professor Klaus Schittkowski for his host, which includes not only valuable academic discussions but also week-end hiking trips.

REFERENCES

[1] J. Barzilai and J. M. Borwein, *Two point step size gradient methods*, IMA J. Numer. Anal., **8** (1988), 141–148.
[2] S. Bellavia, C. Cartis, N. I. M. Gould, B. Morini and Ph. L. Toint, *Convergence of a regularized Euclidean residual algorithm for nonlinear least-squares*, SIAM J. Numer. Anal., **48** (2010), 1–29.
[3] L. Bergamaschi, I. Moret and G. Zilli, *Inexact quasi-Newton methods for sparse systems of nonlinear equations*, Future Generation Computer Systems, **18** (2001), 41–53.
[4] E. G. Birgin, N. Krejic and J. M. Martinez, *Globally convergent inexact quasi-Newton methods for solving nonlinear systems*, Numerical Algorithms, **32** (2003), 249–260.
[5] A. Bouaricha and J. J. Moré, *Impact of Partial Separability on Large-Scale Optimization*, Computational Optimization and Applications, **7** (1997), 27–40.
[6] M. H. Cheng, "Quasi-Newton Type Methods for Solving Large-Scale Problems," Ph.D. thesis, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, 2010.
[7] M. H. Cheng and Y. H. Dai, *Sparse two-sided rank-one updates for nonlinear equations*, Science in China, Ser. A., **53** (2010), 1–10.
[8] A. R. Conn, N. J. M. Gould and Ph. L. Toint, "Trust-Region Methods," MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2000.
[9] J. E. Dennis Jr. and J. J. Moré, *Quasi-Newton methods, motivation and theory*, SIAM Review, **19** (1977), 46–89.
[10] J. E. Dennis and R. B. Schnabel, "Numerical Methods for Unconstrained Optimization and Nonlinear Equations," SIAM, Philadelphia, 1993.
[11] J. Y. Fan and Y. X. Yuan, *On the quadratic convergence of the Levenberg-Marquardt method without nonsingularity assumption,* Computing, **74** (2005), 23–39.
[12] J. Y. Fan and Y. X. Yuan, *Regularized Newton method with correction for monotone nonlinear equations and its application*, Report, AMSS, CAS, Beijing, 2010.
[13] R. Fletcher, *A model algorithm for composite NDO problem*, Math. Program., **17** (1982), 67–76.
[14] R. Fletcher, "Practical Methods of Optimization," Second Edition, John Wiley and Sons, 1987.
[15] R. Fletcher and C. Xu, *Hybrid methods for nonlinear least squares*, IMA J. Numerical Analysis, **7** (1987), 371–389.
[16] G. H. Golub and V. Pereyra, *The differentiation of pseudo-inverses and nonlinear least squares whose variables separable*, SIAM J. Numer. Anal, **10** (1973), 413–432.
[17] G. H. Golub and V. Pereyra, *Separable nonlinear least squares: the variable projection method and applications*, Inverse Problems, **19** (2003), 1–26.
[18] G. H. Golub and C. F. Van Loan, "Matrix Computations," 3rd Edition, Johns Hopkins University Press, Baltimore and London, 1996.

[19] N. I. M. Gould, D. Orban and Ph. L. Toint, *Numerical methods for large-scale nonlinear optimization*, Acta Numerica, 299–361, 2005.
[20] N. I. M. Gould and Ph. L. Toint, *FILTRANE, a Fortran 95 filter-trust-region package for solving nonlinear least-squares and nonlinear feasibility problems*, ACM Transactions on Mathematical Software (TOMS), **33** (2007), 3–25.
[21] A. Griewank, "Evaluating Derivatives. Principles and Techniques of Algorithmic Differentiation," Frontiers in Applied Mathematics, Vol 19, SIAM, Philadelphia, PA, 2000.
[22] A. Griewank and A. Walther, *On constrained optimization by adjoint based quasi-Newton methods*, Optimizaiton Methods and Software, **17** (2002), 869–889.
[23] L. Kaufman, *A variable projection method for solving separable nonlinear least squares problems*, BIT, **15** (1975), 49–57.
[24] C.T. Kelley, "Iterative Methods for Linear and Nonlinear Equations," SIAM, Philadelphia, 1995.
[25] C. T. Kelley, "Solving Nonlinear Equations with Newton's Method," Fundamentals of Algorithms Series, SIAM, Philadelphia, 2003.
[26] C. Kanzow, N. Yamashita and M. Fukushima, *Levenberg-Marquardt methods with storng local convergence properties for solving nonlinear equations with convex constraints*, J. Comp. Appl. Math., **173** (2005), 321–343.
[27] W. La Cruz and M. Raydan, *Nonmonotone spectral methods for large-scale nonlinear sysstems*, Optimization Methods and Software, **18** (2003), 583–599.
[28] W. La Cruz, J. M. Martinez and M. Raydan, *Spectral residual method without gradient information for solving large-scale nonlinear systems of equations*, Math. Comp., **75** (2006), 1429–1448.
[29] D. H. Li and M. Fukushima, *A globally and superlinearly convergent Gauss-Newton-based BFGS method for symmetric nonlinear equations*, SIAM J. Numer. Anal., **37** (1999), 152–172.
[30] D. H. Li and M. Fukushima, *A derivative-free line search and global convergence of Broyden-like method for nonlinear equations*, Optimizaiton Methods and Software, **13** (2000), 181–201.
[31] X. Liu and Y. X. Yuan, *On the separable nonlinear least squares problems*, J. Comput. Math., **26** (2008), 390–403.
[32] J. M. Martinez, *A quasi-Newton method with modification of one column per iteration*, Computing, **33** (1984), 353–362.
[33] E. Mizutani and J. W. Demmel, *On structure-exploiting trust-region regularized nonlinear least squares algorithms for neural-network learning*, Neural Networks, **16** (2003), 745–753.
[34] J. J. Moré, *The Levenberg-Marquardt algorithm: implementation and theory*, in "Lecture Notes in Mathematics 630: Numerical Analysis"(eds. G.A. Watson), Springer-Verlag, Berlin, 1978, pp. 105–116.
[35] J. J. Moré, *Recent developments in algorithms and software for trust region methods*, in "Mathematical Programming: The State of the Art (Springer-Verlag, Berlin, 1983)"(eds. A. Bachem, M. Grötschel and B. Korte), pp. 258–287.
[36] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Compute., **4** (1983), 553–572.
[37] Yu. Nesterov, *Modified Gauss-Newton scheme with worst-case grarantees for global performance*, Optimization Methods and Sofware, **22** (2007), 469–483.
[38] J. Nocedal and S. J. Wright, "Numerical Optimization," Springer, New York, 1999.
[39] J. M. Ortega and W. C. Rheinboldt, "Iterative Solution of Nonlinear Equations in Several Variables," Academic Press, New York and London, 1970.
[40] M. J. D. Powell, *A new algorithm for unconstrained optimization*, in "Nonlinear Programming"(eds. J.B. Rosen, O.L. Mangasarian and K. Ritter), pp. 31–65, Academic Press, New York, 1970.
[41] M. J. D. Powell, *A method for minimizing a sum of squares of non-linear functions without calculating derivatives*, The Computer J., **7** (1965), 303–307.
[42] M. J. D. Powell, *A hybrid method for nonlinear equations*, in "Numerical Methods for Nonlinear Algebraic Equations"(eds. P. Robinowtiz), Gordon and Breach, London, 1970, pp. 87–114.
[43] M. J. D. Powell, *A fast algorithm for nonlinearly constrained optimization calculations*, in "Numerical Analysis"(eds. G.A. Watson), Springer, Berlin, 1978, pp. 144–157.
[44] M. J. D. Powell and Y. X. Yuan, *Conditions for superlinear convergence in l_1 and l_∞ solutions of overdetermined nonlinear equations*, IMA J. Numerical Analysis, **4** (1984), 241–251.

[45] A. Ruhe and P. Å. Wedin, *Algorithms for separable nonlinear least squares problems*, SIAM Review, **22** (1980), 318–337.
[46] Y. Saad, "Iterative Methods for Sparse Linear Systems," SIAM, New York, 2003.
[47] S. Schlenkrich, A. Griewank and A. Walther, *On the local convergence of adjoint Broyden methods*, Math. Program., **121** (2010), 221–247.
[48] K. Schittkowski, *Solving nonlinear least squares problems by a general purpose SQP-method*, in "Trends in Mathematical Optimization"(eds. K.H. Hoffmann, J.-B. Hiriart-Urruty, C. Lemarechal, J. Zowe), International Series of Numerical Mathematics, Vol. 84, Birkhäuser, Boston, 1988, pp. 295–309.
[49] K. Schittkowski, "Numerical Data Fitting in Dynamical Systems," Applied Optimization Vol. 77, Kluwer Academic Publishers, Dordrecht, 2002.
[50] L. K. Schubert, *Modification of a quasi-Newton method for nonlinear equations with a sparse Jacobian*, Math. Comp., **24** (1970), 27–30.
[51] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numerical Analysis, **20** (1983), 626–637.
[52] J. Stoer and Y. X. Yuan, *A subspace study on conjugate gradient algorithms*, Z. Angew. Math. Mech., **75** (1995), 69–77.
[53] W. Y. Sun and Y. X. Yuan, "Optimization Theory and Mehtods: Nonlinear Programming," Springer Series on Optimization and Its Application, Vol. 1, Springer, 2006.
[54] Ph. L. Toint, *On sparse and sysmmetric matrix updating subject to a linear equation*, Mathematics of Computation, **31** (1977), 954–961.
[55] Ph. L. Toint, *Towards an efficient sparsity exploiting Newton method for minimization*, in "Sparse Matrices and Their Uses"(eds. I. Duff), Academic Press, 1981, pp. 57–88.
[56] Ph. L. Toint, *On large scale nonlinear least squares calculations*, SIAM J. Sci. Stat. Comput., **8** (1987), 416–435.
[57] Z. H. Wang and Y. X. Yuan, *A subspace implementation of quasi-Newton trust region methods for unconstrained optimization*, Numerische Mathematik, **104** (2006), 241–269.
[58] N. Yamashita and M. Fukushima, *On the rate of convergence of the Levenberg-Marquardt method* , Computing, (Supplement) **15** (2001), 237–249.
[59] Y. X. Yuan, *Trust region algorithms for nonlinear equations*, Information, **1** (1998), 7–20.
[60] Y. X. Yuan, *On the truncated conjugate gradient method,* Math. Program., **87** (2000), 561–573.
[61] Y. X. Yuan, *Subspace techniques for nonlinear optimization*, in "Some Topics in Industrial and Applied Mathematics" (eds. R. Jeltsch, D.Q. Li and I. H. Sloan), (Series in Contemporary Applied Mathematics CAM 8) Higher Education Press. Beijing, 2007, pp. 206–218.
[62] Y. X. Yuan, *Subspace methods for large scale nonlinear equations and nonlinear least squares*, Optimization and Engineering, **10** (2009), 207–218.
[63] H. C. Zhang, A. R. Conn and K. Scheinberg, *A derivative-free algorithm for the least-squares minimization*, `http://www.math.lsu.edu/~hozhang/zhcpaper.html`.
[64] H. C. Zhang, A. R. Conn and K. Scheinberg, *On the local convergence of a derivative-free algorithm for least-squares minimization*, `http://www.math.lsu.edu/~hozhang/zhcpaper.html`.

Received June 2010; revised September 2010.

E-mail address: `yyx@lsec.cc.ac.cn`