

Article

Not peer-reviewed version

A Simplified Algorithm for a Full-Rank Update Quasi-Newton Method

[Peter Berzi](#) *

Posted Date: 30 December 2024

doi: 10.20944/preprints202412.2511.v1

Keywords: quasi-Newton methods; least-squares solution; multi-variable nonlinear equations; full-rank Jacobian approximate update; convergence; efficiency



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

A Simplified Algorithm for a Full-Rank Update Quasi-Newton Method

Peter Berzi 

Óbuda University, Bécsi út 96/B, 1034 Budapest, Hungary; berzi.peter@uni-obuda.hu

Abstract: One of the most common ideas for finding the zero of a nonlinear function is to replace it by a series of suitably chosen linear functions, for which the zeros can easily be determined and the sequence of zeros approximate the zero of the nonlinear function. Widely used classic methods are Newton's method and a large family of quasi-Newton methods (secant, Broyden's, discretized Newton, Steffensen's,...). This strategy can be called "linearization" and such methods may be called "linearization methods". An efficient linearization method for solving a system of nonlinear equations was developed which showed good stability and convergence properties. It uses an unconventional and simple strategy to improve the performance of classic methods by full-rank update of the Jacobian approximates. It can be considered both as a discretized Newton's method or as a quasi-Newton method with full-rank update of the Jacobian approximates. A solution to the secant equation presented in [2] was based on the Wolfe-Popper procedure [29,42]. The secant equation was splitted into two equations by introducing an auxiliary variable. A simplified algorithm is given in this paper for the full-rank update procedure described in [2]. It directly solves the secant equation with the pseudo-inverse of the Jacobian approximate matrix. Numerical examples are shown for demonstration purpose. The convergence and efficiency of the suggested method are discussed and compared with the convergence and efficiency of classic linearization methods.

Keywords: quasi-Newton methods; least-squares solution; multi-variable nonlinear equations; full-rank Jacobian approximate update; convergence; efficiency

1. Introduction

A mathematical model $\phi(\omega, x)$ with some parameters $x = [x_i]$ ($i = 1, \dots, n$) is constructed for an observed system which gives observable response $D = [D_j]$ sampled at locations ω_j ($j = 1, \dots, m$) ($m > n$) to an observable external effect. It is assumed that the simulated system response $\phi(\omega, x)$ is sensitive to the perturbations Δx_i of all parameters. The n adjustable parameters x of the mathematical model are determined so that the distance

$$l_q = \|f(x)\|_q = \|\phi(\omega, x) - D\|_q \quad (1.1)$$

between the observed and the simulated system responses ($[D_j]$ and $\phi(\omega, x)$) is minimized, where $l_q : \mathbb{R}^n \rightarrow \mathbb{R}$ is a q norm

$$l_q = \left(\sum_{j=1}^m |f_j(x)|^q \right)^{\frac{1}{q}} \quad (1.2)$$

If $q = 2$ is chosen, then we get the Euclidean or least-squares norm

$$l_2 = \left(\sum_{j=1}^m |f_j(x)|^2 \right)^{\frac{1}{2}} \quad (1.3)$$

that is well computable and safely usable for a great class of problems. Then minimizing the distance 1.1 leads to the problem of minimizing the least squares norm

$$l_2 = \|f(x)\|_2 = \|\phi(\omega, x) - D\|_2 \quad (1.4)$$

($x \in \mathbb{R}^n$, $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ and $n < m$). The minimum value of this norm is zero, but for “real life” problems it is not reachable in most cases due to modeling inaccuracies and measurement noises. The suggested procedure gives a least-squares solution to the overdetermined system of nonlinear equations

$$f(x) = 0 \quad (1.5)$$

where the solution x^* minimizes the least squares norm 1.4 of the nonlinear residual function

$$f(x) = \phi(\omega, x) - D \quad (1.6)$$

Least squares minimizations are successively made within the actual iteration steps. The basic concept of solving Equation 1.5 is that the function $f(x)$ is “linearized” by repeatedly replacing it with a linear function $y_p = y(x_p)$ as

$$f(x) \simeq y_p = f_p + J_p(x - x_p) \quad (1.7)$$

for the Newton’s method, where $f_p = f(x_p)$ is the function value, $J_p = J(x_p)$ is the Jacobian matrix of the function $f(x)$ (p is iteration counter) and

$$f(x) \simeq y_p = f_p + S_p(x - x_p) \quad (1.8)$$

for the quasi-Newton methods, where $S_p = S(x_p)$ is the finite-difference Jacobian approximate of the function $f(x)$. Then the nonlinear problem is solved by solving a series of linear problems

$$y_p = 0 \quad (1.9)$$

and it follows that the new approximate x_{p+1} to the solution x^* can be given from Equations

$$J_p \Delta x_p = -f_p \quad (1.10)$$

(Newton’s method) and

$$S_p \Delta x_p = -f_p \quad (1.11)$$

(quasi-Newton methods) where the iteration stepsize is

$$\Delta x_p = x_{p+1} - x_p \quad (1.12)$$

in both cases.

The system of simultaneous multi-variable nonlinear Equations 1.5 can be solved by the Newton’s method when the derivatives J_p of $f(x)$ are available analytically and a new iterate x_{p+1} can be determined. In many cases, explicit formulas for the function $f(x)$ are not available and the Jacobian J_p can only be approximated. The partial derivatives of the analytic Jacobian may be replaced by suitable finite-difference quotients (discretized Newton’s iteration [12,25]) with properly chosen step sizes. For most problems, Newton’s method using analytic derivatives and Newton’s method using properly chosen divided differences are virtually indistinguishable [12]. However, the determination of the finite-difference stepsizes is not clearly defined. The suggested full-rank update procedure helps to overcome this deficit.

Let S_p be a full column rank matrix (all column vectors of matrix S_p are linearly independent). As $n < m$ (there are more rows than columns), S_p is not invertible and the solution to the overdetermined system of Equation 1.11 doesn't exist. However, $S_p^T S_p$ is invertible and the pseudo-inverse

$$S_p^+ = \left(S_p^T S_p \right)^{-1} S_p^T \quad (1.13)$$

of matrix S_p is a unique left inverse which allows finding a least-squares approximate solution \tilde{x} to Equation 1.11. The pseudo-inverse can be determined in different ways (e.g. rank factorization [14], singular value decomposition [31]). Singular value decomposition is a widely used technique, and it has an advantage, that it gives solution even if S_p is singular (not a full column rank matrix). Matrix S_p can be factorized as

$$S_p = U_p \Sigma_p V_p^T \quad (1.14)$$

where U_p ($m \times n$ matrix) and V_p ($n \times n$ matrix) are orthogonal matrices with orthonormal column eigenvectors, and

$$\Sigma_p = \text{diag}(\sigma_{i,p}) \quad (1.15)$$

($i = 1, \dots, n$) with $\sigma_{i,p}$ singular values of matrix S_p and $\sigma_1 \geq \dots \geq \sigma_n$. Then the least square solution to Equation 1.11 exists and the iteration stepsize is

$$\Delta x_p = -S_p^+ f_p \quad (1.16)$$

where

$$S_p^+ = V_p \Sigma_p^{-1} U_p^T \quad (1.17)$$

is the pseudoinverse of S_p and

$$\Sigma_p^{-1} = \text{diag} \left(\frac{1}{\sigma_{i,p}} \right) \quad (1.18)$$

The pseudoinverse $[\dots]^+$ gives a unique least squares solution to Equation 1.11 for which the $q = 2$ norm (Euclidean norm) 1.4 of the residual function 1.6 is minimum. If the rank of matrix S_p is less than n (the column vectors of S_p are not linearly independent and some singular values are zero), then unique solution to Equation 1.11 doesn't exist. The spectral condition number

$$\kappa(S_p) = \frac{\sigma_1}{\sigma_n} \geq 1 \quad (1.19)$$

of matrix S_p measures the linear dependency of the column vectors. If they are linearly independent (non of the singular values are zero) and the condition number κ is not much larger than one, then the matrix is well-conditioned. This is a desirable situation through the whole iteration process. Then the column vectors of matrix S_p are in "general positions" and the Ortega-Rheinboldt condition [25] satisfies. If $\kappa \gg 1$ then the matrix is ill-conditioned and the solution Δx_p (iteration stepsize) to the Equation 1.11 may be sensitive to small changes in S_p or f_p .

Classic quasi-Newton linearization methods generate a sequence of improved iterates x_p ($p = 0, 1, 2, \dots$) so that the next approximate x_{p+1} to the solution x^* is determined on the basis of rank-1 update S_{p+1} of the Jacobian approximate S_p . Such linearization update is not unique if only a single new approximate x_{p+1} exists. In single variable case ($n = 1$), there are two possible updates (secant lines) from which the selection is not obvious (see details in Section 3). In multi variable case, only rank-1 update is possible if one single new approximate x_{p+1} exists.

The solution to Equation 1.11 presented in [2] and applied to the identification of physically nonlinear dynamic systems [3–5] was based on the Wolfe-Popper procedure [29,42]. The secant equation 1.11 was splitted into two equations by introducing an auxiliary variable. The simplified solution presented in Section 4 directly solves Equation 1.11 with the pseudo-inverse 1.13 of matrix S_p . The simplified algorithm given in Section 5 gives an unconventional and simple strategy to improve the

performance of Newton and quasi-Newton iterations for the solution of system of nonlinear equations 1.5. It can be considered both as a discretized Newton's method or as a quasi-Newton method with full-rank update of the Jacobian approximates.

The rate of convergence α of the classic secant method equals to the golden section ratio φ ($\alpha = \varphi \simeq 1.618$) for simple root in single variable case. The suggested method gives an additional new independent approximate x_{p+1}^B to the solution x^* , then rank- n linearization update can be made with the classic approximate x_{p+1} (notated as x_{p+1}^A in the followings) and the suggested new approximate x_{p+1}^B . The results of single- and multi-variable numerical examples are given in Section 6. The efficiency of the proposed method is discussed in Section 7 and it is compared with other classic rank-one update and line-search methods on the basis of available test data. Concluding remarks are summarized in Section 8.

2. Notations

Vectors and matrices are denoted by bold-face letters. Subscripts refer to components of vectors and matrices, superscripts A and B refer to interpolation base points. Notations A and B are introduced to be able to clearly distinguish between the two new approximates x^A and x^B . Vectors and matrices may also be given by their general elements. Δ refers to a difference of two elements. x and X denotes unknown quantities, f and F denotes function values and matrices. t and T denotes multiplier scalars and scaling transformation matrices. e , ε and E denotes approximate error, p is iteration counter, α is convergence rate, ε^* is termination criterion. n is the number of unknowns, m is the number of function values, i , j and k and are running indexes of matrix columns and rows. Superscripts S and TS refer to the traditional Secant-method and to the suggested full-rank update method (T-Secant) receptively.

3. Linearization Methods

The origin of the secant method can be traced back to ancient time to the "rule of double false position" method, described in the 18th century B.C. on the Egyptian Rhind Papyrus [28] and it predates Newton's method by more than 3000 years. It is also well-known that the local convergence rate of the classic Newton's and the classic secant methods for simple roots are quadratic and super-linear respectively [8,11,26,38]. Weerakson and Fernando's third order Newton variant [40] involves evaluation of a function value and two derivatives, similarly like in Traub's suggestion [38]. Secant method variations with improved convergence have been reported in literature [13,15–17,23,27,33,34,40]. Mueller's method [24] uses a quadratic approximation. A class of variations employs two function value and a derivative evaluation [37]. Third order methods are obtained by a second derivative evaluation, such as Halley's [23] and the "super Halley" [1] method. Further improvements with higher degree polynomial approximations are proposed by Chen [9], Kanwar [17], Zhang [43] and Wang [39].

3.1. Single-Variable Case

The zero x^* of a scalar nonlinear function $f(x)$ ($x \rightarrow f(x)$, $x \in \mathbb{R}^1$, $f: \mathbb{R}^1 \rightarrow \mathbb{R}^1$) has to be determined, where

$$f(x^*) = 0 \quad (3.1)$$

Linearization means that $f(x)$ is locally replaced by a linear function $y(x)$ ($x \rightarrow y(x)$, $x \in \mathbb{R}^1$, $y: \mathbb{R}^1 \rightarrow \mathbb{R}^1$) (secant- or tangent line) as shown on Figure 1 (Left) and operations are made on the linear function $y_p(x)$ in the p^{th} iteration. $y_p(x)$ may be defined through two points $A_p(x_p^A, f_p^A)$ and $B_p(x_p^B, f_p^B)$ of the nonlinear function $f(x)$, where $f_p^A = f(x_p^A)$ and $f_p^B = f(x_p^B)$ are function values, or through one point $A_p(x_p^A, f_p^A)$ and by the "slope"

$$S_p = \frac{\Delta f_p}{\Delta x_p} \quad (3.2)$$

of the secant line $y_p(x)$, where $\Delta x_p = x_p^B - x_p^A$ and $\Delta f_p = f_p^B - f_p^A$ are differences, or by the “slope”

$$S_p = f'(x_p^A) = \frac{df}{dx} \quad (3.3)$$

at x_p^A of the tangent line $y_p(x)$, where f' is the derivative function of $f(x)$. The successive local replacement of the nonlinear function $f(x)$ by a secant- or tangent line $y_p(x)$ gives a simple and efficient numerical root finding procedure. It follows from the condition

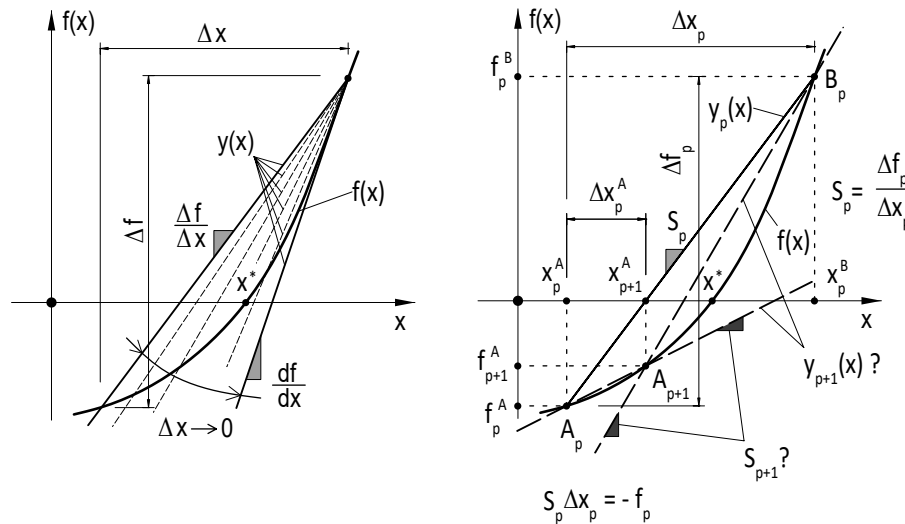


Figure 1. Left : Linearization of a nonlinear function, Right : Classic secant method

$$y_p(x_{p+1}^A) = 0 \quad (3.4)$$

that the zero

$$x_{p+1}^A = x_p^A - \frac{f_p^A}{S_p} \quad (3.5)$$

of the linear function $y_p(x)$ approximates the zero x^* of the nonlinear function $f(x)$ and the new point $A_{p+1}(x_{p+1}^A, f_{p+1}^A)$ can be determined for the next iteration as shown on Figure 1 (Right). With the iteration step length

$$\Delta x_p^A = x_{p+1}^A - x_p^A \quad (3.6)$$

from point A_p , Equation 3.5 can be rewritten as

$$S_p \Delta x_p^A = -f_p^A \quad (3.7)$$

that is called secant equation or we may also call it “linearization” equation as it also includes Newton’s method as a special case (tangent line). The secant based procedure has an advantage that it doesn’t need the calculation of function derivatives, it only uses function values and the order of asymptotic convergence is super-linear with convergence rate $\alpha^S \cong 1.618...$. However, the “slope” of the updated secant line $y_{p+1}(x)$ can be determined in two ways from the secant (quasi-Newton) conditions as

$$S_{p+1} \Delta x_p^A = f_{p+1}^A - f_p^A \quad (3.8)$$

or

$$S_{p+1} \Delta x_p^B = f_{p+1}^A - f_p^B \quad (3.9)$$

where

$$\Delta x_p^B = x_{p+1}^A - x_p^B \quad (3.10)$$

is the iteration step length from point B_p . The decision is far not obvious, as shown on Figure 1 (Right). The tangent line based iteration (Newton's method) has an advantage that the "slope"

$$S_{p+1} = f'(x_{p+1}^A) \quad (3.11)$$

of the updated tangent line is always well-defined. The iteration then continues with updated secant- or tangent line $y_{p+1}(x)$.

3.2. Multi-Variable Case

The scalar linearization procedure can be extended to multi-dimensions. The zero \mathbf{x}^* of a nonlinear vector-valued function

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{bmatrix} \quad (3.12)$$

of n variables

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad (3.13)$$

($\mathbf{x} \rightarrow \mathbf{f}(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $n \leq m$) has to be determined, where

$$\mathbf{f}(\mathbf{x}^*) = 0 \quad (3.14)$$

Then $\mathbf{f}(\mathbf{x})$ is locally replaced by an n -dimensional hyperplane

$$\mathbf{y}(\mathbf{x})(\mathbf{x} \rightarrow \mathbf{y}(\mathbf{x}), \mathbf{x} \in \mathbb{R}^n, \mathbf{y} : \mathbb{R}^n \rightarrow \mathbb{R}^m) \quad (3.15)$$

($n \leq m$) and operations are made on the hyperplane $\mathbf{y}_p(\mathbf{x})$ in the p^{th} iteration. $\mathbf{y}_p(\mathbf{x})$ may be defined through $n+1$ points $A_p(\mathbf{x}_p^A, \mathbf{f}_p^A)$ and $B_{k,p}(\mathbf{x}_{k,p}^B, \mathbf{f}_{k,p}^B)$ ($k = 1, \dots, n$) of the nonlinear function $\mathbf{f}(\mathbf{x})$, where $\mathbf{f}_p^A = \mathbf{f}(\mathbf{x}_p^A)$ and $\mathbf{f}_{k,p}^B = \mathbf{f}(\mathbf{x}_{k,p}^B)$ ($k = 1, \dots, n$) are function values, or by one point $A_p(\mathbf{x}_p^A, \mathbf{f}_p^A)$ and by the "slope" (divided differences)

$$\mathbf{S}_p = \Delta \mathbf{F}_p \Delta \mathbf{X}_p^{-1} \quad (3.16)$$

($i = 1, \dots, n$), ($j = 1, \dots, m$), ($k = 1, \dots, n$) of the n -dimensional secant hyperplane $\mathbf{y}_p(\mathbf{x})$, where $\Delta \mathbf{x}_{k,p} = \mathbf{x}_{k,p}^B - \mathbf{x}_p^A$ and $\Delta \mathbf{f}_{k,p} = \mathbf{f}_{k,p}^B - \mathbf{f}_p^A$ are difference vectors with n and m components respectively, and

$$\Delta \mathbf{F}_p = [\Delta \mathbf{f}_{k,p}] = [\mathbf{f}_{k,p}^B - \mathbf{f}_p^A] = \begin{bmatrix} \Delta f_{1,1,p} & \cdots & \Delta f_{n,1,p} \\ \vdots & \ddots & \vdots \\ \Delta f_{1,m,p} & \cdots & \Delta f_{n,m,p} \end{bmatrix} \quad (3.17)$$

$$\Delta \mathbf{X}_p = [\Delta \mathbf{x}_{k,p}] = [\mathbf{x}_{k,p}^B - \mathbf{x}_p^A] = \begin{bmatrix} \Delta x_{1,1,p} & \cdots & \Delta x_{n,1,p} \\ \vdots & \vdots & \vdots \\ \Delta x_{1,n,p} & \cdots & \Delta x_{n,n,p} \end{bmatrix} \quad (3.18)$$

The n -dimensional tangent hyperplane $\mathbf{y}_p(\mathbf{x})$ through point $A_p(\mathbf{x}_p^A, \mathbf{f}_p^A)$ may be defined by the “slope”

$$\mathbf{S}_p = \left[\frac{\partial f_{k,j,p}}{\partial x_{k,i,p}} \right] = \begin{bmatrix} \frac{\partial f_{1,1,p}}{\partial x_{1,p}} & \cdots & \frac{\partial f_{n,1,p}}{\partial x_{n,p}} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_{1,m,p}}{\partial x_{1,p}} & \cdots & \frac{\partial f_{n,m,p}}{\partial x_{n,p}} \end{bmatrix} \quad (3.19)$$

that corresponds to the Jacobian matrix \mathbf{J}_p of the nonlinear vector-valued function $\mathbf{f}(\mathbf{x})$ and accordingly, Definition 3.16 corresponds to the approximation of the Jacobian matrix. It follows from the condition

$$\mathbf{y}_p(\mathbf{x}_{p+1}^A) = \mathbf{0} \quad (3.20)$$

that the zero

$$\mathbf{x}_{p+1}^A = \mathbf{x}_p^A - \mathbf{S}_p^+ \mathbf{f}_p^A \quad (3.21)$$

of the n -dimensional hyperplane $\mathbf{y}(\mathbf{x})$ approximates the zero \mathbf{x}^* of the nonlinear vector-valued function $\mathbf{f}(\mathbf{x})$, where $[\cdot]^+$ stands for the pseudo-inverse. Then the i^{th} element of the new approximate \mathbf{x}_{p+1}^A in the p^{th} iteration will be

$$x_{i,p+1}^A = x_{i,p}^A - \sum_{j=1}^m (S_{i,j,p}^+ f_{j,p}^A) \quad (3.22)$$

($i = 1, \dots, n$), ($j = 1, \dots, m$). With the iteration stepsize

$$\Delta \mathbf{x}_p^A = \mathbf{x}_{p+1}^A - \mathbf{x}_p^A = -\mathbf{S}_p^+ \mathbf{f}_p^A = - \left[\left(\sum_{j=1}^m (S_{i,j,p}^+ f_{j,p}^A) \right)_i \right] \quad (3.23)$$

Equation 3.21 can be rewritten as

$$\mathbf{S}_p \Delta \mathbf{x}_p^A = -\mathbf{f}_p^A \quad (3.24)$$

that is called “secant equation” or “linearization equation” in multi-variable case. With the new approximate \mathbf{x}_{p+1}^A and with the corresponding function value \mathbf{f}_{p+1}^A , an updated Jacobian or Jacobian approximate \mathbf{S}_{p+1} has to be determined. Table 1 summarizes the basic equations of the above detailed linearization methods.

If all partial derivatives of the function $\mathbf{f}(\mathbf{x})$ is known, then the Jacobian update can easily be calculated. Newton’s method is one of the most widely used algorithm with very attractive theoretical and practical properties and with some limitations. The computational costs of Newton’s method is high, since the Jacobian \mathbf{J}_p and the solution to the linear system 3.24 must be computed at each iteration. It is well-known that the local convergence of Newton’s method is q -quadratic if the initial trial approximate \mathbf{x}_0 is close enough to the solution \mathbf{x}^* , if $\mathbf{J}(\mathbf{x}^*)$ is non-singular and if $\mathbf{J}(\mathbf{x})$ satisfies the Lipschitz condition

$$\|\mathbf{J}(\mathbf{x}) - \mathbf{J}(\mathbf{x}^*)\| \leq L \|\mathbf{x} - \mathbf{x}^*\| \quad (3.25)$$

for all \mathbf{x} close enough to \mathbf{x}^* .

However, in many cases, the function $\mathbf{f}(\mathbf{x})$ is not an analytical function, the partial derivatives are not known or difficult to evaluate and Newton’s method cannot be applied. Quasi-Newton methods are widely used for solving systems of nonlinear equations when the Jacobian is not known or difficult to determine. The Jacobian is approximated by divided differences 3.16, and the system of nonlinear equations 1.5 is solved by repeatedly solving systems of linear equations 3.24 providing the new approximates \mathbf{x}_{p+1}^A .

Table 1. Linearization method's basic equations (single- and multi-variable cases)

| | Single-variable ($m = n = 1$) | Multi-variable ($m \geq n > 1$) | Equations |
|---|--|--|-----------|
| 1 | $\Delta x_p = x_p^B - x_p^A$ | $\Delta \mathbf{X}_p = \begin{bmatrix} \Delta x_{1,1,p} & \cdots & \Delta x_{n,1,p} \\ \vdots & \vdots & \vdots \\ \Delta x_{1,n,p} & \cdots & \Delta x_{n,n,p} \end{bmatrix}$ | 3.18 |
| 2 | $\Delta f_p = f_p^B - f_p^A$ | $\Delta \mathbf{F}_p = \begin{bmatrix} \Delta f_{p,1,1} & \cdots & \Delta f_{p,n,1} \\ \vdots & \ddots & \vdots \\ \Delta f_{p,1,m} & \cdots & \Delta f_{p,n,m} \end{bmatrix}$ | 3.17 |
| 3 | $S_p = \frac{\Delta f_p}{\Delta x_p}$ | $S_p = \Delta \mathbf{F}_p \Delta \mathbf{X}_p^{-1}$ | 3.2, 3.16 |
| 4 | $x_{p+1}^A = x_p^A - \frac{f_p^A}{S_p}$ | $\mathbf{x}_{p+1}^A = \mathbf{x}_p^A - S_p^+ f_p^A$ | 3.5, 3.21 |
| 5 | $S_{p+1} \Delta x_p^A = f_{p+1}^A - f_p^A$ | $S_{p+1} \Delta \mathbf{x}_p^A = f_{p+1}^A - f_p^A$ | 3.8, 3.26 |

The Jacobian approximate 3.16 should be updated according to the fundamental equation of the quasi-Newton methods ("quasi-Newton condition" or "secant condition")

$$S_{p+1} \Delta \mathbf{x}_p^A = f_{p+1}^A - f_p^A \quad (3.26)$$

for all $p = 0, 1, 2, \dots$. However, the condition 3.26 doesn't uniquely specify the update S_{p+1} , so the iterative procedure is not well-defined and further constraints are needed. Different methods offer their own specific solution, but new quasi-Newton approximate \mathbf{x}_{p+1} will never allow full-rank update of the Jacobian approximate S_{p+1} (Equation 3.26 is an underdetermined system of m linear equations with $m \cdot n$ unknowns). A large family of methods are available with different additional conditions or assumptions. Martinez [21] has been made a thorough survey on the family of practical quasi-Newton methods.

The partial derivatives of the Jacobian matrix may be replaced by suitable difference quotients (discretized Newton iteration, see [12,25])

$$\left[\frac{\Delta \mathbf{f}}{\Delta \mathbf{x}} \right] = \left[\frac{f_j(\mathbf{x} + \Delta x_k \mathbf{d}^k) - f_j(\mathbf{x})}{\Delta x_k} \right] = \left[\frac{\Delta f_j(\mathbf{x})}{\Delta x_k} \right] \quad (3.27)$$

($k = 1, \dots, n$), ($j = 1, \dots, m$) with n additional function value evaluations, where \mathbf{d}^k is the k^{th} Cartesian unit vector. However, it is difficult to choose the stepsize $\Delta \mathbf{x}$. If any Δx_k is too large, then Expression 3.27 can be a bad approximation to the Jacobian so the iteration converges much more slowly, if it converges at all. On the other hand, if any Δx_k is too small, then $\Delta f_j(\mathbf{x}) \simeq 0$, and cancellations can occur which reduces the accuracy of the difference quotients 3.27 (see [35]). Another modification is the inexact-Newton approach, when the nonlinear equation is solved by an iterative linear solver (see [6,10,22]). Wolfe [42] and Popper [29] suggested column update of matrix 3.17.

One of the most widely used Broyden-formula [7] makes a rank-one update for the Jacobian approximate as

$$S_{p+1} = S_p + Q_p = S_p + \frac{\Delta f_p^A - S_p \Delta \mathbf{x}_p^A}{\|\Delta \mathbf{x}_p^A\|^2} [\Delta \mathbf{x}_p^A]^T \quad (3.28)$$

where Q_p is a rank-one matrix and by using the Sherman-Morrison formula, the inverse Jacobian approximate update is given as

$$S_{p+1}^+ = S_p^+ + \frac{\Delta \mathbf{x}_p^A - S_p^+ \Delta f_p^A}{[\Delta \mathbf{x}_p^A]^T S_p^+ \Delta f_p^A} [\Delta \mathbf{x}_p^A]^T S_p^+ \quad (3.29)$$

Broyden's secant condition 3.26 can be re-written as

$$S_p \Delta x_p^A + Q_p \Delta x_p^A = f_{p+1}^A - f_p^A \quad (3.30)$$

and with the secant Equation 3.24 we have

$$Q_p \Delta x_p^A = f_{p+1}^A \quad (3.31)$$

As Q_p is a rank-one matrix, then this equation has infinitely large number of solutions (the left nullspace of matrix Q_p is an $n - 1$ dimensional vector space [36]) and the Ortega-Rheinboldt [25] condition will not be satisfied (the column vectors of matrix S_p should be linearly independent and have to be "in general position" through the whole iteration process) and it may result an ill-conditioned Broyden update S_{p+1} .

4. T-Secant Method

A numerical procedure has been developed for solving an overdetermined system of nonlinear equations 1.5. It can be considered both as a discretized Newton's method or as a quasi-Newton method with full-rank update of the Jacobian approximates. The suggested full-rank update procedure ("T-Secant" method [2]) provides an unconventional and simple strategy to improve the performance of quasi-Newton iterations. It is based on the classic secant linearization, that was completed with a new independent approximate x_{p+1}^B . The secant equation was modified by a suitably chosen non-uniform scaling transformation and a new independent approximate x_{p+1}^B was determined from the modified secant equation. The solution to the secant equation presented in [2] was based on the Wolfe-Popper procedure [29,42] so that the secant equation was splitted into two equations by introducing an auxiliary variable. The simplified algorithm, presented in the paper, directly solves the secant equation by using the pseudo-inverse of the transformed Jacobian approximate.

4.1. Single-Variable Case

The suggested procedure uses the information on the improvement of the classic secant procedure and gives a new approximate x_{p+1}^B in the vicinity of the classic secant approximate x_{p+1}^A as follows. Given two initial approximates x_p^A and x_p^B with function values f_p^A and f_p^B and the new approximate x_{p+1}^A with function value f_{p+1}^A is known from the solution of the classic secant Equation 3.7. An independent new approximate

$$x_{p+1}^B = x_{p+1}^A + \Delta x_{p+1} \quad (4.1)$$

is planned to be determined in the vicinity of the classic secant approximate x_{p+1}^A . Let the ratio

$$t_p^f = \frac{f_{p+1}^A}{f_p^A} \quad (4.2)$$

$(f_p^A \neq 0)$ of the function value improvement and the ratio

$$t_p^x = \frac{x_{p+1}^B - x_{p+1}^A}{x_{p+1}^A - x_p^A} = \frac{\Delta x_{p+1}}{\Delta x_p^A} \quad (4.3)$$

$(\Delta x_p^A \neq 0)$ of the desired iteration stepsize change be defined. The single variable secant equation 3.7 is then modified as

$$S_{Tp} \Delta x_p^A = -f_p^A \quad (4.4)$$

where

$$S_{Tp} = \frac{t_p^f}{t_p^x} S_p \quad (4.5)$$

Then the new approximate x_{p+1}^B can be expressed from Equation 4.4

$$t_p^f \frac{x_{p+1}^A - x_p^A}{x_{p+1}^B - x_{p+1}^A} S_p \Delta x_p^A = -f_p^A \quad (4.6)$$

After re-arrangement

$$x_{p+1}^A - x_{p+1}^B = \frac{t_p^f}{f_p^A} S_p (\Delta x_p^A)^2 \quad (4.7)$$

and with the secant Equation 3.7

$$x_{p+1}^B = x_{p+1}^A - \frac{t_p^f}{f_p^A} S_p (\Delta x_p^A)^2 = x_{p+1}^A + t_p^f \Delta x_p^A \quad (4.8)$$

the desired new iteration stepsize

$$\Delta x_{p+1} = t_p^f \Delta x_p^A \quad (4.9)$$

can be given. The suggested procedure can also be applied to the Newton's method (T-Newton method). Then the "slope" S_p corresponds to the derivative f' of function $f(x)$.

The geometrical representation of the suggested method can be derived as follows ([2]). Let the new approximate x_{p+1}^B be the zero of a function $z_p(x)$:

$$z_p(x_{p+1}^B) = 0 \quad (4.10)$$

Then by replacing x_{p+1}^B with x in Equation 4.8 gives the function

$$z_p(x) = \frac{t_p^f S_p (\Delta x_p^A)^2}{x - x_{p+1}^A} + f_p^A = 0 \quad (4.11)$$

with zero at

$$x = x_{p+1}^B \quad (4.12)$$

Equation 4.11 is a hyperbolic function with vertical and horizontal asymptotes x_{p+1}^A and f_p^A , and its root x_{p+1}^B will be in the vicinity of x_{p+1}^A in "appropriate distance" that is regulated by the function value f_{p+1}^A (see Figure 2). This virtue of the suggested procedure ensures an automatic mechanism for having the actual approximates x_p^A and x_p^B in general positions through the whole iteration process providing stable and efficient numerical performance. The suggested method's performance is demonstrated with the results of numerical tests on test functions in Section 6.

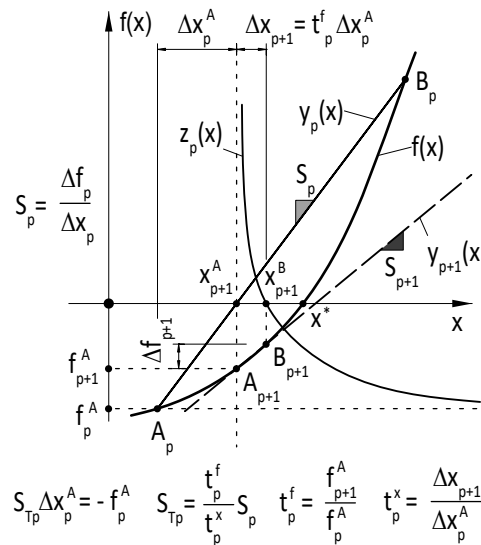


Figure 2. Suggested full-rank update method (T-secant)

4.2. Multi-Variable Case

Let two independent approximates $\mathbf{x}_p^A = [\mathbf{x}_{k,p}^A]$ and $\mathbf{x}_p^B = [\mathbf{x}_{k,p}^B]$ to the zero \mathbf{x}^* of the nonlinear vector-valued function $\mathbf{f}(\mathbf{x})$ be given in the p^{th} iteration ($k = 1, \dots, n$). Let the series of n approximates $\mathbf{x}_{k,p}^B$ be constructed by individually increment the elements $\mathbf{x}_{k,p}^A$ of the approximate \mathbf{x}_p^A by an increment

$$\Delta x_{k,p} = x_{k,p}^B - x_{k,p}^A \quad (4.13)$$

as

$$\mathbf{x}_{k,p}^B = \mathbf{x}_p^A + \Delta x_{k,p} \mathbf{d}^k \quad (4.14)$$

where d^k is the k^{th} Cartesian unit vector. It follows from this special construction of the approximates $x_{k,p}^B$, that $x_{k,i,p}^B - x_{k,p}^A = 0$ for $i \neq k$ and $x_{k,i,p}^B - x_{k,p}^A = \Delta x_{k,p}$ for $i = k$ and matrix 3.18 will be a diagonal matrix :

$$\Delta \mathbf{X}_p = [\Delta \mathbf{x}_{k,p}] = [\mathbf{x}_{k,p}^B - \mathbf{x}_p^A] = \begin{bmatrix} \Delta x_{1,p} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \Delta x_{n,p} \end{bmatrix} = \text{diag}(\Delta x_{k,p}) \quad (4.15)$$

Let the ratios

$$T_p^F = \text{diag}(t_{j,p}^F) = \text{diag}\left(\frac{f_{j,p+1}^A}{f_{j,p}^A}\right) \quad (4.16)$$

$(f_{j,p}^A \neq 0)$ of the function value improvements ($j = 1, \dots, m$) and the ratios

$$T_p^X = \text{diag}(t_{i,p}^X) = \text{diag}\left(\frac{x_{i,p+1}^B - x_{i,p+1}^A}{x_{i,p+1}^A - x_{i,p}^A}\right) = \text{diag}\left(\frac{\Delta x_{i,p+1}}{\Delta x_{i,p}^A}\right) \quad (4.17)$$

$(\Delta x_{i,p}^A \neq 0)$ of the desired iteration stepsize changes $(i = 1, \dots, n)$ be defined similarly like in the single variable case. Let the secant equation 3.24 be modified as

$$S_{Tp} \Delta x_p^A = -f_p^A \quad (4.18)$$

where

$$\mathbf{S}_{T_p} = \mathbf{T}_p^F \mathbf{S}_p (\mathbf{T}_p^X)^{-1} = \text{diag}(t_{j,p}^F) \left[\frac{\Delta f_{k,j,p}}{\Delta x_{i,p}} \right] \text{diag} \left(\frac{1}{t_{i,p}^X} \right) = \left[\frac{t_{j,p}^F}{t_{i,p}^X} \frac{\Delta f_{k,j,p}}{\Delta x_{i,p}} \right] \quad (4.19)$$

$(i = 1, \dots, n), (j = 1, \dots, m), (k = 1, \dots, n)$ or in explicit form :

$$S_{Tp} = \begin{bmatrix} \frac{f_{1,p+1}^A}{f_{1,p}^A} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{f_{m,p+1}^A}{f_{m,p}^A} \end{bmatrix} \begin{bmatrix} \frac{\Delta f_{1,1,p}}{\Delta x_{1,p}} & \dots & \frac{\Delta f_{1,n,p}}{\Delta x_{n,p}} \\ \vdots & \vdots & \vdots \\ \frac{\Delta f_{1,m,p}}{\Delta x_{1,p}} & \dots & \frac{\Delta f_{n,m,p}}{\Delta x_{n,p}} \end{bmatrix} \begin{bmatrix} \frac{\Delta x_{1,p}^A}{\Delta x_{1,p+1}} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{\Delta x_{n,p}^A}{\Delta x_{n,p+1}} \end{bmatrix} \quad (4.20)$$

and

$$S_{Tp}^+ = T_p^X S_p^+ (T_p^F)^{-1} \quad (4.21)$$

is the pseudoinverse of S_p . Then Equation 4.18 can be re-written as

$$\Delta x_p^A = -S_{Tp}^+ f_p^A \quad (4.22)$$

Then the i^{th} element $x_{i,p+1}^B$ of the new approximate x_{p+1}^B in the p^{th} iteration will be

$$x_{i,p+1}^B = x_{i,p+1}^A + \Delta x_{i,p+1} = x_{i,p+1}^A - \frac{(\Delta x_{i,p}^A)^2}{\sum_{j=1}^m \left(S_{i,j,p}^+ \frac{f_{j,p}^A}{t_{j,p}^F} \right)} \quad (4.23)$$

where $t_{p,j}^F \neq 0, (j = 1, \dots, m)$ and $(i = 1, \dots, n)$. Let the ratios

$$\mu_{i,p} = \frac{\sum_{j=1}^m \left(S_{i,j,p}^+ f_{j,p}^A \right)}{\sum_{j=1}^m \left(S_{i,j,p}^+ \frac{f_{j,p}^A}{t_{j,p}^F} \right)} = \frac{-\Delta x_{i,p}^A}{\sum_{j=1}^m \left(S_{i,j,p}^+ \frac{f_{j,p}^A}{t_{j,p}^F} \right)} \quad (4.24)$$

be introduced . By using Equation 3.23, the new iteration stepsize can be expressed from Equation 4.23 as

$$\Delta x_{i,p+1} = x_{i,p+1}^B - x_{i,p+1}^A = - \frac{(\Delta x_{i,p}^A)^2}{\sum_{j=1}^m \left(S_{i,j,p}^+ \frac{f_{j,p}^A}{t_{j,p}^F} \right)} = \mu_{i,p} \Delta x_{i,p}^A \quad (4.25)$$

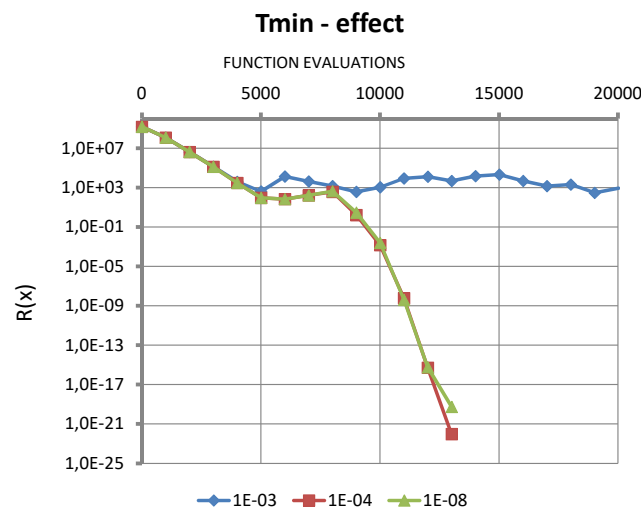
Table 2 summarizes the basic equations of the above detailed multi-variable update method. The suggested procedure can also be applied to the Newton's method (T-Newton method). Then matrix S_p corresponds to the Jacobian matrix J_p of function $f(x)$. The function value improvement parameter $t_{j,p}^F$ ($j = 1, \dots, m$) in Equation 4.24, defined as 4.16 has a key role in the suggested iteration process and the absolute value of the denominator $\left| \frac{f_{j,p}^A}{t_{j,p}^F} \right|$ has to be low bounded with a pre-given value T_{min} to avoid division by zero. When the values $\left| \frac{f_{j,p}^A}{t_{j,p}^F} \right|$ tend to zero with increasing iteration counter p , then

$$\mu_{i,p} \rightarrow T_{min} \quad (4.26)$$

Figure 3 shows the effect of iteration parameter T_{min} on the iteration process ($N = 1000$). It is clear that too high value of T_{min} has negative effect on the iteration efficiency. It can be seen on the example that a value 0.001 causes the iteration process failure, and there is almost no effect with lower values $0.0001 \leq T_{min}$.

Table 2. Basic equations of classic and suggested multi-variable update methods

| Classic secant method | Suggested update method | Equations |
|---|--|------------|
| 1 | $\mathbf{x}_{k,p}^B = \mathbf{x}_p^A + \Delta \mathbf{x}_{k,p} \mathbf{d}^k$ | 4.14 |
| 2 $\Delta \mathbf{X}_p = [\Delta \mathbf{x}_{k,p}]$ | $\Delta \mathbf{X}_p = \text{diag}(\Delta \mathbf{x}_{k,p})$ | 3.18, 4.15 |
| 3 $\Delta \mathbf{F}_p = [\mathbf{f}_{k,p}^B - \mathbf{f}_p^A]$ | $\Delta \mathbf{F}_p = [\mathbf{f}_{k,p}^B - \mathbf{f}_p^A]$ | 3.17 |
| 4 | $\mathbf{T}_p^X = \text{diag}(t_{i,p}^X) = \text{diag}\left(\frac{\Delta x_{i,p+1}}{\Delta x_{i,p}^A}\right)$ | 4.17 |
| 5 | $\mathbf{T}_p^F = \text{diag}(t_{j,p}^F) = \text{diag}\left(\frac{f_{j,p+1}^A}{f_{j,p}^A}\right)$ | 4.16 |
| 6 $\mathbf{S}_p = \Delta \mathbf{F}_p \Delta \mathbf{X}_p^{-1}$ | $\mathbf{S}_{Tp} = \mathbf{T}_p^F \mathbf{S}_p (\mathbf{T}_p^X)^{-1} = \left[\frac{t_{j,p}^F \Delta f_{k,j,p}}{t_{i,p}^X \Delta x_{i,p}^A} \right]$ | 3.16, 4.19 |
| 7 $\mathbf{S}_p \Delta \mathbf{x}_p^A = -\mathbf{f}_p^A$ | $\mathbf{S}_{Tp} \Delta \mathbf{x}_p^A = -\mathbf{f}_p^A$ | 3.24, 4.18 |
| 8 $\Delta \mathbf{x}_p^A = -\mathbf{S}_p^+ \mathbf{f}_p^A$ | $\Delta \mathbf{x}_p^A = -\mathbf{S}_{Tp}^+ \mathbf{f}_p^A$ | 3.23, 4.22 |
| 9 $\mathbf{x}_{i,p+1}^A = \mathbf{x}_{i,p}^A - \sum_{j=1}^m (\mathbf{S}_{i,j,p}^+ f_{j,p}^A)$ | $\mathbf{x}_{i,p+1}^B = \mathbf{x}_{i,p+1}^A - \frac{(\Delta x_{i,p}^A)^2}{\sum_{j=1}^m (\mathbf{S}_{i,j,p}^+ \frac{f_{j,p}^A}{t_{j,p}^F})}$ | 3.22, 4.23 |
| 10 $\mathbf{x}_{i,p+1}^B = \mathbf{x}_{i,p}^B$ | $\mu_{i,p} = \frac{\sum_{j=1}^m (\mathbf{S}_{i,j,p}^+ f_{j,p}^A)}{\sum_{j=1}^m (\mathbf{S}_{i,j,p}^+ \frac{f_{j,p}^A}{t_{j,p}^F})}$ | 4.24 |
| 11 $\Delta \mathbf{x}_{i,p+1} = \mathbf{x}_{i,p+1}^B - \mathbf{x}_{i,p+1}^A$ | $\Delta \mathbf{x}_{i,p+1} = \mu_{i,p} \Delta \mathbf{x}_{i,p}^A$ | 4.25 |

**Figure 3.** Effect of the iteration parameter T_{min} on the iteration process ($N = 1000$)

The suggested procedure may also be repeated without using the classic secant approximates and without updating the previously known Jacobian approximate \mathbf{S}_p as

$$\Delta \mathbf{x}_{i,p+1+h} = \mathbf{x}_{i,p+1+h}^B - \mathbf{x}_{i,p+1+h}^A = \frac{-\left(\sum_{j=1}^m (\mathbf{S}_{i,j,p}^+ f_{j,p+h}^A)\right)^2}{\sum_{j=1}^m \left(\mathbf{S}_{i,j,p}^+ \frac{f_{j,p+h}^A}{t_{j,p+h}^F}\right)} \quad (4.27)$$

where

$$t_{j,p+h}^F = \frac{f_{j,p+1+h}^A}{f_{j,p+h}^A} \quad (4.28)$$

($h = 1, \dots$) until the new approximates

$$x_{i,p+1+h}^B = x_{i,p+1+h}^A + \Delta x_{i,p+1+h} \quad (4.29)$$

($i = 1, \dots, n$) are sufficiently improved. A numerical example is shown in Table 3 with test function

$$f(x) = x^3 - 2x - 5 \quad (4.30)$$

for demonstration purpose. The results indicate linear convergence with convergence rate $\alpha = 1$. α is the computed convergence rate, N_f is the number of function value evaluations and L is the mean convergence rate, suggested by Broyden [7] in Table 3.

Table 3. T-Secant iteration with constant Jacobian approximate S_p (Equation 4.27)

| h | x_{p+h}^A | x_{p+h}^B | Δx_{p+h}^A | t_{p+h}^f | Δx_{p+1+h} | $ e_{p+1+h}^A $ | α_{p+h}^T | N_f | L |
|-----|-------------|-------------|--------------------|-------------|--------------------|---------------------|------------------|-------|------|
| 0 | 1.700 | 2.000 | 0.421 | -0.085 | 0.036 | $2.6 \cdot 10^{-2}$ | | 2 | 1.23 |
| 1 | 2.121 | 2.156 | -0.036 | -0.359 | -0.013 | $2.2 \cdot 10^{-2}$ | | 4 | 0.87 |
| 2 | 2.085 | 2.072 | 0.013 | -0.342 | 0.0044 | $7.6 \cdot 10^{-3}$ | 1.08 | 6 | 0.76 |
| 3 | 2.098 | 2.102 | -0.0044 | -0.348 | -0.0015 | $2.7 \cdot 10^{-3}$ | 0.97 | 8 | 0.70 |
| 4 | 2.093 | 2.092 | 0.0015 | -0.346 | 0.00053 | $9.2 \cdot 10^{-4}$ | 1.01 | 10 | 0.67 |
| 5 | 2.0949 | 2.0955 | -0.00053 | -0.347 | -0.00018 | $3.2 \cdot 10^{-4}$ | 0.997 | 12 | 0.65 |
| 6 | 2.0944 | 2.0942 | 0.00018 | -0.346 | 0.000063 | $1.1 \cdot 10^{-4}$ | 1.001 | 14 | 0.63 |
| 7 | 2.0946 | 2.0947 | -0.000063 | -0.346 | 0.000022 | $3.8 \cdot 10^{-5}$ | 0.9997 | 16 | 0.62 |

5. Algorithm

Let ε^* be the error bound for termination criterion and if x^* is known then

$$e_p^A = x_p^A - x^* \quad (5.1)$$

is the error vector of approximate x_p^A in the p^{th} iteration with elements $e_{i,p}^A$ ($i = 1, \dots, n$). Let the error norm

$$\varepsilon_p = \frac{\|e_p^A\|_2}{n} = \frac{\sqrt{\sum_{i=1}^n (e_{i,p}^A)^2}}{n} \quad (5.2)$$

be defined, where $\|\cdot\|_2$ is Euclidean norm and let the iteration be terminated when

$$\varepsilon_p < \varepsilon^* \quad (5.3)$$

holds. Choose T_{min} as lower bound for $|t_{j,p}^F|$ ($j = 1, \dots, m$) and let f_{min} be a lower bound for $|f_{j,p}^A|$ ($j = 1, \dots, m$). Let $p = 0$ and let the approximate x_p^A and the difference vector Δx_p be given. Calculate the corresponding function values f_p^A and assure that $f_{min} < |f_{j,0}^A|$ ($j = 1, \dots, m$). Iteration constants f_{min} and T_{min} are necessary to avoid division by zero and to avoid computed values be near the numerical precision.

- Step 1 : Generate a set of n additional approximates $x_{k,p}^B$ (Equation 4.14) and evaluate function values $f_{k,p}^B$ ($k = 1, \dots, n$). Assure that $f_{min} < |f_{k,j,p}^B|$ ($j = 1, \dots, m$).
- Step 2 (Classic secant update method) : Construct the Jacobian approximate matrix S_p , determine its pseudo-inverse S_p^+ [31], calculate x_{p+1}^A from Equation 3.21 and ε_p from Equation 5.2.
- Step 3 : If $\varepsilon_p < \varepsilon^*$ then terminate iteration, else continue with Step 4.

- Step 4 (suggested update method) : Calculate f_{p+1}^A (assure that $f_{min} < |f_{j,p}^A|$), T_p^F from Equation 4.16 and μ_i from Equation 4.24 . Let $T_{min} < |t_{j,p}^F|$ and determine x_{p+1}^B from Equation 4.25.
- Step 5 : Continue iteration from Step 1 with $p = p + 1$, x_{p+1}^A and x_{p+1}^B .

If p_{max} is the number of necessary iterations for satisfying the termination criterion $\varepsilon_p < \varepsilon^*$ and n is the number of unknowns to be determined, then the suggested update method needs $n + 1$ function evaluations in each iterations and altogether

$$N_f = p_{max}(n + 1) \quad (5.4)$$

function evaluations to reach the desired termination criterion. p_{max} is depending on many circumstances such as the nature of the function $f(x)$, termination criteria (ε^* or others), the distance of the initial approximate x^A from the solution x^* and from the iteration constants f_{min} and T_{min} .

6. Numerical Tests Results

6.1. Single Variable Test Function

A numerical example is given with a single-variable test function

$$f(x) = \cos(x) - x \quad (6.1)$$

with root $x^* \cong 0.73908513...$ The results of the classic secant iteration process are shown on Table 4 and Figure 4. Iterations were made with initial approximates $x_0^A = -2.0$ and $x_0^B = 2.0$ ($p = 0$) providing $f_0^A = 1.584$. The first secant approximate $x_1^A = -0.416...$ is found as the zero of the first secant line $y_0(x)$ providing $f_1^A = 1.331...$ ($p = 1$). The next iteration ($p = 2$) will continue with approximate $x_2^A = 0.442.....$ and with $f_2^A = 0.462...$.

The results of the suggested iteration process are shown on Table 5 and Figure 5. Iterations were made with the same initial approximates as in case of the classic secant iteration and the first secant approximate $x_1^A = -0.416...$ is found. The first T-secant approximate $x_1^B = 0.915...$ is given as the zero of the first hyperbola function $z_0(x)$ (Figure 5, left). Iteration then goes on with approximates $x_1^A = -0.416...$ and $x_1^B = 0.915...$ providing $f_1^A = 1.331...$ ($p = 1$), and new approximates $x_2^A = 0.667...$ and $x_2^B = 0.764...$ as the zeros of the second secant and hyperbola functions $y_1(x)$ and $z_1(x)$ receptively (Figure 5, left). The next iteration ($p = 2$) will then continue with approximates $x_2^A = 0.667.....$ and $x_2^B = 0.764...$ and with $f_2^A = 0.119...$ and gives $x_3^A = 0.7387.....$ and $x_3^B = 0.7391...$ (Figure 5, right).

Table 4. Classic secant iterations with test function 6.1 (see Figure 4)

| p | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|--------------|--------|--------|--------|--------|---------------------|---------------------|----------------------|
| x_p^A | -2.000 | 2.000 | -0.416 | 0.442 | 0.898 | 0.7279 | 0.7387 |
| Δx_p | 4.000 | -2.416 | 0.858 | 0.456 | -0.170 | 0.011 | $3.6 \cdot 10^{-4}$ |
| f_p^A | 1.584 | -2.416 | 1.331 | 0.462 | -0.275 | 0.019 | $6.1 \cdot 10^{-4}$ |
| x_{p+1}^A | -0.416 | 0.442 | 0.898 | 0.728 | 0.7387 | 0.739086 | 0.739085 |
| e_{p+1}^A | -1.155 | -0.297 | 0.159 | -0.011 | -0.0004 | $9.1 \cdot 10^{-7}$ | $7.3 \cdot 10^{-11}$ |
| α^S | | | 0.113 | 15.5 | 0.459 | 4.249 | 1.292 |
| f_{p+1}^A | 1.331 | 0.462 | -0.275 | 0.019 | $6.1 \cdot 10^{-4}$ | $1.5 \cdot 10^{-6}$ | $1.2 \cdot 10^{-10}$ |

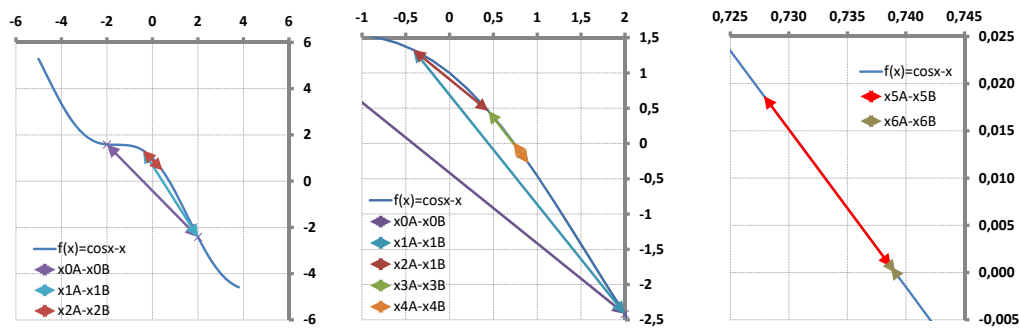


Figure 4. Classic secant iterations with test function 6.1 (see Table 4). (Left : $x_1^A = -0.416 \dots$ is the root of $y_0(x)$, then $x_2^A = 0.442 \dots$ is the root of $y_1(x)$. Middle : $x_3^A = 0.898 \dots$ is the root of $y_2(x)$ and $x_4^A = 0.728 \dots$ is the root of $y_3(x)$ Right : $x_5^A = 0.7387 \dots$ is the root of $y_4(x)$ and $x_6^A = 0.739086 \dots$ is the root of $y_5(x)$)

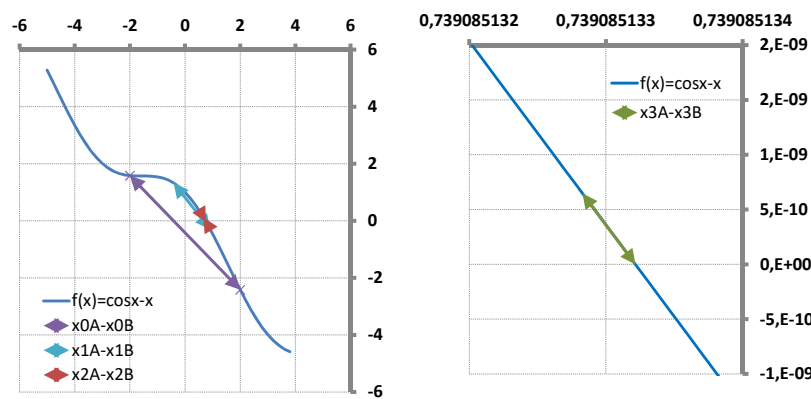


Figure 5. T-Secant iterations with test function 6.1 (see Table 5) (Left : $x_1^A = -0.416 \dots$ is the root of $y_0(x)$, $x_1^B = 0.915 \dots$ is the root of $z_0(x)$, then $x_2^A = 0.667 \dots$ is the root of $y_1(x)$, $x_2^B = 0.764 \dots$ is the root of $z_1(x)$. Right : $x_3^A = 0.7387 \dots$ is the root of $y_2(x)$, $x_3^B = 0.7391 \dots$ is the root of $z_2(x)$)

Table 5. T-Secant iterations with test function 6.1 (see Figure 5)

| p | 0 | 1 | 2 | 3 | 4 |
|---------------|--------|--------|----------------------|----------------------|----------------------|
| x_p^A | -2.000 | -0.416 | 0.6668 | 0.7387 | 0.7390851328 |
| Δx_p | 4.000 | 1.331 | 0.0968 | $4.1 \cdot 10^{-4}$ | $3.9 \cdot 10^{-10}$ |
| f_p^A | 1.584 | 1.331 | 0.1190 | $6.7 \cdot 10^{-4}$ | $6.5 \cdot 10^{-10}$ |
| x_{p+1}^A | -0.416 | 0.667 | 0.7387 | 0.7390851328 | 0.7390851332 |
| e_{p+1}^A | -1.155 | -0.072 | $-4.0 \cdot 10^{-4}$ | $3.8 \cdot 10^{-10}$ | |
| α^{TS} | | | 3.210 | 1.874 | 2.668 |
| f_{p+1}^A | 1.331 | 0.119 | $6.7 \cdot 10^{-4}$ | $6.5 \cdot 10^{-10}$ | |
| t_p^F | 0.840 | 0.089 | 0.0057 | $9.6 \cdot 10^{-7}$ | |
| x_{p+1}^B | 0.915 | 0.764 | 0.7391 | 0.7390851332 | |

6.2. Solution of an Inverse Problem

An example is given for an imaginary modeling problem. Modeling of systems on the basis of observed or specified system responses plays a key role in scientific research. A central problem of modeling is to fit a structured mathematical model to available data so that the values of the unknown model parameters are determined providing that the simulated data is as near to the available data as possible. A system response can generally be represented by a "curve" in a two (or three) dimensional space and can be simulated by a computer program. Parameter identification corresponds to minimize the distance between observed and simulated system responses. Parameter identification problems can

generally be formulated as non-linear least squares problems so that some unknown model parameters are determined providing minimal deviation between observed and simulated system responses. Solving non-linear least squares problems is often a difficult task especially if the number of unknowns is high.

A rational system under investigation produces measurable response to a known measurable external effect. A mathematical model with n unknown parameters (p_1, \dots, p_n) simulates the behavior of the system providing response to the same external effect as acted on the system during observation. Let, the observed and the simulated system responses be represented by two dimensional curves and sampled in k discrete points with coordinates x_i and y_i ($i = 1, \dots, k$). The coordinates of a synthetic observed response were generated as

$$x(z) = (a + b) \cos(z + ff) - c \cos\left(\frac{a+b}{b} z + ff\right) + x_0 \quad (6.2)$$

$$y(z) = (a + b) \sin(z + ff) - c \sin\left(\frac{a+b}{b} z + ff\right) + y_0 \quad (6.3)$$

(an epicycloid) with parameters $x_0 = 10, y_0 = 8, a = 4, b = 2, c = 3.5$ and $ff = 1$ and with $0 \leq z \leq 2\pi$. The parameters

$$p = \begin{bmatrix} x_0 & y_0 & a & b & c \end{bmatrix} \quad (6.4)$$

were considered to be unknown, and the system response was simulated with initial approximate

$$p_0 = \begin{bmatrix} x_0 & y_0 & a & b & c \end{bmatrix} = \begin{bmatrix} 8 & 11 & 3.5 & 2.5 & 3 \end{bmatrix} \quad (6.5)$$

Distance between observed and simulated system responses was defined for arbitrary two-dimensional curves. This definition can be extended to n dimensions making wide range of parameter identification problems possible to formulate. The value of the defined distance is dimensionless and it expresses the ratio of area between system response graphs in normalized coordinate system to the area of a rectangle with unit area.

The distance between the observed and the simulated system responses was quantified by area $D(p)$ between the graphs of system responses in a normalized coordinate system. Partial area $r_j(p)$ was defined as the area of a quadrangle among the $(j-1)^{th}$ and j^{th} division points on the system response graphs ($j = 1, \dots, m$) as shown on Figure 6. These division points were selected equidistantly along system response graphs.

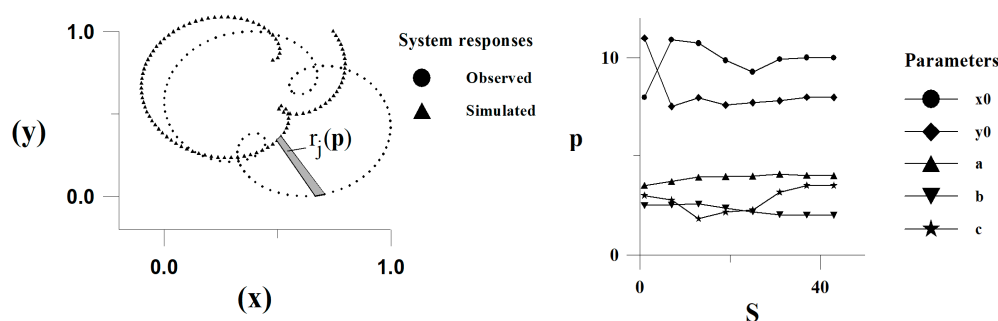


Figure 6. Definition of the distance between system responses (left) and variation of parameters through iterations (right)

Thus, parameter identification problems can be formulated as solving a system of nonlinear equations

$$r(p) = 0 \quad (6.6)$$

where $p \rightarrow r(p)$, $p \in \mathbb{R}^5$, $r: \mathbb{R}^5 \rightarrow \mathbb{R}^m$ and r is residual vector with components r_j ($j = 1, \dots, m$). The unknown parameters p of an epicycloid 6.2 and 6.3 were determined according to the suggested update algorithm. The variation of parameters p through iterations and the observed and the simulated system responses for the initial approximate are shown on Figure ?? and 6. The observed and simulated system response pairs are shown on Figure 7 7, 13, 19, 25, 31 and 43 simulations (S).

The results of numerical tests with different initial approximates showed that the optimal solutions were not reached in all cases. The convergence was especially sensitive for the initial values of parameters a , b and c . Several local optimal solutions were detected as shown on Figure 8.

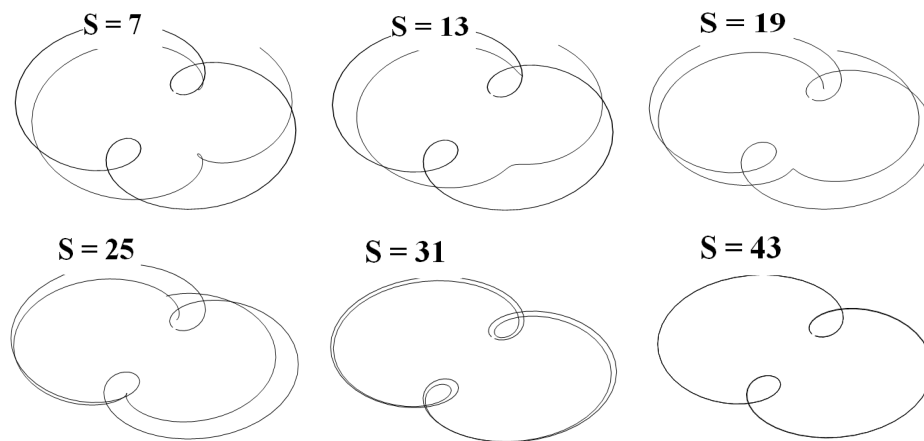


Figure 7. Observed and simulated system response pairs after “S” function value evaluations

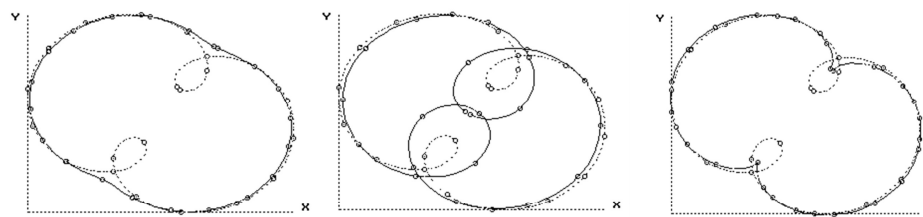


Figure 8. Local optimal solutions

7. Efficiency

The efficiency of an algorithm for the solution of nonlinear equations is thoroughly discussed by Traub [38] as follows. Let f_i be the order of the iteration sequence such that for the approximate errors $e_i = x_i - x^*$, there exists a nonzero constant C (asymptotic error constant) for which

$$\frac{|e_{i+1}|}{|e_i|^{f_i}} \rightarrow C \quad (7.1)$$

A natural measure of the information used by an algorithm is the “informational usage” d , which is defined as the number of new pieces of information (values of the function and its derivatives) required per iteration (called “horner” by Ostrowski [26]). Then the efficiency of the algorithm within one iteration can be measured by the “informational efficiency”

$$EFF = \frac{f_i}{d} \quad (7.2)$$

An alternative definition of efficiency is

$$*EFF = f_i^{\frac{1}{d}} \quad (7.3)$$

called “efficiency index” by Ostrowski [26]. Another measure of efficiency, called “computational efficiency” takes into account the “cost” of calculating different derivatives. The concept of informa-

tional efficiency (*EFF*) and efficiency index (**EFF*) doesn't take into account the cost of evaluating *f* and its derivatives, nor does they take into account the total number of pieces of information needed to achieve a certain accuracy in the root of the function. If *f* is composed of elementary functions, then the derivatives are also composed of elementary functions, thus the cost of evaluating the derivatives is merely the cost of combining the elementary functions.

Broyden [7] suggested the mean convergence rate

$$L = \frac{1}{N_f} \ln \frac{R(x_0^A)}{R(x_{p_{max}}^A)} \tag{7.4}$$

as a measure of efficiency of an algorithm for solving a particular problem, where *N_f* is the total number of function evaluations, *x₀^A* is the initial approximate, *x_{p_{max}}^A* is the last approximate to the solution *x** when the termination criteria is satisfied after *p_{max}* iterations. *R(x)* is the Euclidean norm of *f(x)*.

Zang's two-step method [43] is similar to King-Werner method [18,41] with convergence rate 2.414... [32]. Chen's [9] and Wang's [39] three-step methods show asymptotic convergence order 2.732.... Chen's [9] method require two function and one derivative evaluations by using a second-order polynomial, proposed in [39] and shows 1.174... Ostrowski's [26] efficiency index. Wang's [39] method require three function and two derivative evaluations shows 1.101... Ostrowski-index. Zhang [43] published a finite difference based secant method with 2.618... asymptotic convergence order. Later Ren [32] showed that Zhang's [43] method has only 2.414... convergence order due to some mistakes in the derivation. Table 6 compares the efficiencies of classic (Rows 1-2 : Secant, Newton) and improved algorithms (Rows 3, 5 : T-Secant, T-Newton with the suggested full-rank update and from references [9,39]), the T-Secant method with constant Jacobian approximate *S_p* (row 4 : TS-const. *S_p*, see data in Table 3), Chen's [9] and Wang's [39] methods.

Very limited data are available to compare the performance of the suggested update method (T-Secant) with other classic methods, especially for large number of unknowns. Efficiency results were given by Broyden [7] for the Rosenbrock function for *N* = 2. The calculated convergence rates for the two Broyden method variants [7], for the Powell's method [30], for the adaptive coordinate descent method [19] and for the Nelder-Mead simplex method [20] were compared with the calculated values for the T-secant method in Table 7 [2]. Rows 1-5 are data from referenced papers, rows 6-8 are T-secant results with the referenced initial approximates and rows 9-15 are calculated data for *N* > 2. If the value of *R(x)* = 0 (for *p_{max}*) is zero, then the mean convergence rates (*L* and *L_N*) are not countable (zero in the denominator). A substitute value 10⁻²⁵ was used when *R(x)* = 0 (for *p_{max}*) in rows 6, 7, 10 and 13.

Table 6. Efficiencies of classic and improved algorithms

| | Method | <i>d</i> | <i>β</i> | <i>EFF</i> [38] | <i>*EFF</i> [26] | <i>L_{max}</i> [7] |
|---|---|----------|-----------------|-----------------|------------------|----------------------------|
| 1 | Secant | 1 | 1.618... | 1.618... | 1.618... | 4.0 |
| 2 | Newton | 2 | 2.0 | 1.0 | 1.414... | 3.0 |
| 3 | T-Secant | 2 | 2.618... | 1.309... | 1.618... | 4.5 |
| 4 | TS - const. <i>S_p</i> | 2 | 1.0 | 0.5 | 1.0 | 0.6 |
| 5 | T-Newton | 3 | 3.0 | 1.0 | 1.442... | 3.0 |
| 6 | Chen [9] | 3 | 1.618... | 0.539... | 1.173... | — |
| 7 | Wang [39] | 5 | 1.618... | 0.323... | 1.101... | — |

Table 7. Calculated values of the mean convergence rates

| | <i>N</i> | Method | $R(x_0^A)$ | $R(x_{p_{max}}^A)$ | p_{max} | N_f | <i>L</i> | L_N |
|----|----------|------------------|------------|--------------------|-----------|-------|--------------|-------|
| 1 | 2 | Broyden 1. [7] | 4.92 | $4.7 E^{-10}$ | - | 59 | 0.391 | 0.78 |
| 2 | 2 | Broyden 2. [7] | 4.92 | $2.6 E^{-10}$ | - | 39 | 0.607 | 1.22 |
| 3 | 2 | Powell [30] | 4.92 | $7.0 E^{-10}$ | - | 151 | 0.150 | 0.30 |
| 4 | 2 | ACD [19] | 130.1 | $1.0 E^{-10}$ | - | 325 | 0.086 | 0.17 |
| 5 | 2 | Nelder-Mead [20] | 2.00 | $1.4 E^{-10}$ | - | 185 | 0.127 | 0.25 |
| 6 | 2 | T-secant [7][30] | 4.92 | $1.0 E^{-25}$ | 3 | 9 | 6.573 | 13.15 |
| 7 | 2 | T-secant [19] | 130.1 | $1.0 E^{-25}$ | 3 | 9 | 6.937 | 13.87 |
| 8 | 2 | T-secant [20] | 2.00 | $6.7 E^{-15}$ | 2 | 6 | 5.556 | 11.11 |
| 9 | 3 | T-secant | 72.72 | $1.4 E^{-14}$ | 5 | 20 | 1.809 | 5.43 |
| 10 | 3 | | 32.47 | $1.0 E^{-25}$ | 4 | 16 | 3.815 | 11.45 |
| 11 | 5 | | 93.53 | $1.3 E^{-14}$ | 8 | 48 | 0.760 | 3.80 |
| 12 | 5 | | 7.19 | $5.9 E^{-14}$ | 4 | 24 | 1.351 | 6.76 |
| 13 | 10 | | 202.6 | $1.0 E^{-25}$ | 14 | 154 | 0.408 | 4.08 |
| 14 | 200 | | 92.78 | $9.0 E^{-15}$ | 10 | 2010 | 0.042 | 8.44 |
| 15 | 1000 | | 212.4 | $3.6 E^{-13}$ | 6 | 6006 | 0.006 | 5.66 |

Results show that the mean convergence rate *L* (Equation 7.4) for *N* = 2 is much higher for the T-secant method ($\simeq 5.5 - 6.9$) than for the other listed methods ($\simeq 0.1 - 0.6$). However it is obvious that the mean convergence rate values decrease rapidly with increasing *N* values (more unknowns need more function evaluations). A modified convergence rate

$$L_N = N * L = \frac{N}{N_f} \ln \frac{R(x_0^A)}{R(x_{p_{max}}^A)}$$

(7.5)

is suggested to use as an “*N*” independent measure of efficiency (see Table 7). The values of *L* and *L_N* are at least 10 times larger for the T-secant method than for the referenced classic methods for *N* = 2 (see Table 7).

The efficiency measures (*L* and *L_N*) are also depending on the initial conditions (distance of the initial approximate from the optimal solution, termination criterion). Results from large number of numerical tests indicate an average *L_N* $\simeq 7.4$ with standard deviation around 3.7 for the T-secant method even for large *N* values.

8. Conclusions

A numerical procedure has been developed for solving an overdetermined system of nonlinear equations 1.5. It can be considered both as a discretized Newton’s method or as a quasi-Newton method with full-rank update of the Jacobian approximates. Quasi-Newton methods are widely used for solving systems of nonlinear equations when the function derivatives (Jacobian) are not known or difficult to determine. The derivatives (Jacobian) are approximated by divided differences 3.16, and the system of nonlinear equations 1.5 is solved by repeatedly solving systems of linear equations 3.24. The new approximate x_{p+1}^A is determined from the classic secant equation and the divided differences are updated according to the secant condition 3.26. However, the secant condition doesn’t uniquely specify the Jacobian approximate, so the update procedure is not well-defined and further constraints are needed. Different methods offer specific update solutions.

The suggested numerical procedure (“T-Secant” method [2]) provides an unconventional and simple strategy to improve the performance of quasi-Newton iterations by full-rank update of the Jacobian approximates. It is based on the classic secant linearization and allows full-rank update of the Jacobian approximates. The classic secant iteration was completed with a new independent

approximate x_{p+1}^B , that was determined from a modified secant equation 4.18. Modification was made by a suitably chosen non-uniform scaling transformation. Scaling was made by the difference quotients of the classic secant function value improvements 4.16 and the difference quotients of the desired new iteration stepsizes and the classic secant stepsizes 4.17. A new independent approximate x_{p+1}^B was determined from the modified secant equation 4.18. The solution to the secant equation presented in [2] was based on the Wolfe-Popper procedure [29,42] so that the secant equation was splitted into two equations by introducing an auxiliary variable. The simplified algorithm, presented in the paper, directly solves the secant equation by using the pseudo-inverse 4.21 of the transformed Jacobian approximate 4.19.

It has been shown that the new T-secant approximate x_{p+1}^B will be in the vicinity of the classic secant approximate x_{p+1}^A if the classic secant iterates converge to the root of the nonlinear function [2]. The Jacobian approximate was then full-rank updated by constructing new divided differences (Jacobian approximates) from the classic secant and from the T-secant approximates.

It was shown that the iterative procedure possesses super-quadratic asymptotic convergence property with convergence rate $\alpha = \varphi + 1 = \varphi^2 \simeq 2.618$ for simple root in single variable case ([2]), where φ is the golden section ratio. The suggested procedure can also be applied to the Newton's method (matrix S_p corresponds to the Jacobian matrix J_p). Numerical test results indicate that the quadratic convergence ($\alpha = 2$) of the Newton's method increases to cubic ($\alpha = 3$).

The efficiency has been studied in multi-variable case and compared with other classic rank-one update and line-search methods on the basis of available data. Results show that its efficiency is considerably better than the efficiency of other classic low-rank update methods. The suggested method's performance was demonstrated by the results of numerical tests with widely used single- and multi-variable benchmark test functions. A Rosenbrock test function was used with up to 1000 variables [2]. The method has also been successfully applied for the solution of different "real life" inverse problems [3–5] on physically nonlinear dynamic systems. Further studies can be made on convergence properties in case of multiple roots and on the efficiency characteristics in cases of more benchmark test functions. The suggested procedure may be used for other applications especially with large number of unknowns and when low number of function evaluations is crucial.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: A considerable part of the research work has been done between years 1988 -1992 at Technical University of Budapest (Hungary), at TNO-BOUW Structural Division (The Netherlands) and at Technical High-school of Lulea (Sweden). The work has been sponsored by the Technical University of Budapest (Hungary), by the Hungarian Academy of Sciences (Hungary), by TNO-BOUW (The Netherlands), by Sandvik Rock Tools (Sweden), by CP Test a/s (Denmark) and by Óbuda University (Hungary). Valuable discussions and personal supports from Géza Petrasovits, György Popper, Peter Middendorp, Rikard Skov, Bengt Lundberg, Mario Martinez and Csaba J. Hegedűs are greatly appreciated

Conflicts of Interest: The author declares no conflict of interest..

References

1. Amat, S., Busquier, S., Gutiérrez, J.M. Geometric constructions of iterative functions to solve nonlinear equations. *J. Comput. Appl. Math.* **2003**, *157*, 197-205.
2. Berzi, P. Convergence and Stability Improvement of Quasi-Newton Methods by Full-Rank Update of the Jacobian Approximates. *MDPI AppliedMath* **2024**, *4*, 143-181.
3. Berzi, P., Beccu, R., Lundberg, B. Identification of a Percussive Drill Rod Joint from its Response to Stress Wave Loading. *International Journal of Impact Engineering*. **1994**, *18* (3), 281-290.
4. Berzi, P. Pile-Soil Interaction due to Static and Dynamic Load. *Proceedings of the 13th International Conference on Soil Mechanics and Foundation Engineering*, **1994**, 1994-01-05, New Delhi, India, pp. 609-612.

5. Berzi, P., Popper, Gy. Evaluation of dynamic load test results on piles. Proceedings of the International Symposium on Identification of Nonlinear Mechanical Systems from Dynamic Tests (Euromech 280), **1991** 1991-10-29, Ecully, France, pp. 121-128.
6. Birgin, E.G., Krejic, N., Martinez, J.M. Globally convergent inexact quasi-Newton methods for solving nonlinear systems. *Numer. Algorithms*. **2003**, 32, 249-260.
7. Broyden, C. G. A class of Methods for Solving Nonlinear Simultaneous Equations. *Mathematics of Computation*. American Mathematical Society, **1965**, 19(10), 577-593.
8. Broyden, C.G., Dennis, J.E., Mor, J.J. On the local and superlinear convergence of quasi-Newton methods. *J. Inst. Math. Appl.* **1973**, 12, 223-245.
9. Chen, L., Ma, Y. A new modified King–Werner method for solving nonlinear equations. *Computers and Mathematics with Applications*. **2011**, 62, 3700-3705.
10. Dembo, R.S., Eisenstat, S.C., Steihaug, T. Inexact Newton methods. *SIAM J. Numer. Anal.* **1971**, 19, 400-408.
11. Dennis, J.E., Mor, J.J. A characterization of superlinear convergence and its application to quasi-Newton methods. *Mathematics and Computation* **1974**, 28, 543-560.
12. Dennis, J.E. Jr., Schnabel, R.B. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Prentice-Hall, Englewood Cliffs, NJ, **1983**.
13. Gerlach, J. Accelerated convergence in Newton's method, *SIAM Rev.* **1994**, 36, 272-276.
14. Hegedus, Cs. Numerical Methods I. ELTE, Faculty of Informatics, Budapest, **2015**.
15. Homeier, H.H.H. On Newton-type methods with cubic convergence. *J. Comput. Appl. Math.* **2005**, 176, 425-432.
16. Jisheng, K., Yitian, L., Xiuhua, W., Third-order modification of Newton's method. *J. Comput. Appl. Math.* **2007**, 205, 1-5.
17. Kanwar, V., Sharma, J.R., Mamta J. A new family of Secant-like method with super-linear convergence. *Appl. Math. Comput.* **2005**, 171, 104-107.
18. King, R.F. Tangent method for nonlinear equations. *Numer. Math.* **1972**, 18, 298-304.
19. Loshchilov, I., Schoenauer, M., Sebag, M. Adaptive Coordinate Descent. Genetic and Evolutionary Computation Conference (GECCO), ACM Press, **2011**, 885-892.
20. Nelder, J., A., Mead, R. A simplex method for function minimization. *Computer Journal*. **1965**, 7 (4), 308-313.
21. Martínez, J., M. Practical quasi-Newton methods for solving nonlinear systems. *Journal of Computational and Applied Mathematics*. **2000**, 124, 97-121.
22. Martinez, J.M., Qi, L. Inexact Newton methods for solving non-smooth equations. *J. Comput. Appl. Math.* **1995**, 60, 127-145.
23. Melman, A. Geometry and convergence of Euler's and Halley's methods, *SIAM Rev.* 39 (4) (1997) 728–735. *SIAM Rev.* **1997**, 39(4), 728-735.
24. Muller, D.E., (1956) A Method for Solving Algebraic Equations Using an Automatic Computer. *Math. Tables and Other Aids to Computation* **1956**, 10, 208-215.
25. Ortega, J.M., Rheinboldt, W.C. Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.
26. Ostrowski, A.M. Solution of Equations and Systems of Equations. Academic Press, New York, **1966**.
27. Özban, A.Y. Some new variants of Newton's method. *Appl. Math. Letter*. **2004**, 17, 677-682.
28. Papakonstantinou, J.M., Tapia, R.A. Origin and evolution of the secant method in one dimension. *The American Mathematical Monthly*. **2013**, 120/6, 500-518.
29. Popper, Gy. Numerical method for least square solving of nonlinear equations. *Periodica Polytechnica*. **1985**, 29 (1-2), 67-69.
30. Powell, M., J., D. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal*. **1964**, 7 (2), 155-162.
31. Press, W.H., Flannery, B.P., Teukolsky, S.A., Wetterling, W.T. Numerical recipes. Cambridge University Press, Cambridge, **1986**.
32. Ren, H., Wu, Q., Bi, W. On convergence of a new secant-like method for solving nonlinear equations. *Appl. Math. Comput.* **2010**, 217, 583-589.
33. Scavo, T.R., Thoo, J.B., On the geometry of Halley's method. *American Math. Monthly*. **1995**, 102, 417-426.
34. Shaw, S., Mukhopadhyay, B. An improved regula falsi method for finding simple roots of nonlinear equations. *Appl. Math. and Computation* **2015**, 254, 370-374.
35. Stoer, J., Bulirsch, R. Introduction to Numerical Analysis. Springer -Verlag, **2002**.
36. Strang, G. Introduction to Linear Algebra. Revised International Edition, Wellesley-Cambridge Press, **2005**.

37. Thukral, R. A New Secant-type method for solving nonlinear equations. *Amer. J. Comput. Appl. Math.* **2018**, *8*(2), 32-36.
38. Traub, J.F. *Iterative Methods for the Solution of Equations*, 1st ed.; Prentice-Hall, Inc: Englewood Cliffs, N.J., **1964**.
39. Wang, X., Kou, J., Gu, C. A new modified secant-like method for solving nonlinear equations. *Comput. Math. Appl.* **2010**, *60*, 1633-1638.
40. Weerakoon, S., Fernando, T.G.I., A variant of Newton's method with accelerated third-order convergence. *Appl. Math. Letter.* **2000**, *13*, 87-93.
41. Werner, W., Über ein Verfahren der Ordnung $1 + \sqrt{2}$ zur Nullstellenbestimmung. *Numer. Math.* **1979**, *32*, 333-342.
42. Wolfe, P. The Secant Method for Simultaneous Nonlinear Equations. *Communications of the ACM.* **1959**, *2*, 12-13.
43. Zhang, H., Li, D-S., Liu, Y-Z. A new method of secant-like for nonlinear equations. *Commun. Nonlinear Sci. numer. Simul.* **2009**, *14*, 2923-2927.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.