

Article

Hybrid Newton-like Inverse Free Algorithms for Solving Nonlinear Equations

Ioannis K. Argyros ^{1,*} , Santhosh George ² , Samundra Regmi ³  and Christopher I. Argyros ⁴¹ Department of Computing and Mathematical Sciences, Cameron University, Lawton, OK 73505, USA² Department of Mathematical & Computational Science, National Institute of Technology Karnataka, Surathkal 575025, India; sgeorge@nitk.edu.in³ Department of Mathematics, University of Houston, Houston, TX 77205, USA; sregmi5@uh.edu⁴ School of Computational Science and Engineering, Georgia Institute of Technology, North Avenue Atlanta, Atlanta, GA 30332, USA; cargyros3@gatech.edu

* Correspondence: iargyros@cameron.edu

Abstract: Iterative algorithms requiring the computationally expensive in general inversion of linear operators are difficult to implement. This is the reason why hybrid Newton-like algorithms without inverses are developed in this paper to solve Banach space-valued nonlinear equations. The inverses of the linear operator are exchanged by a finite sum of fixed linear operators. Two types of convergence analysis are presented for these algorithms: the semilocal and the local. The Fréchet derivative of the operator on the equation is controlled by a majorant function. The semi-local analysis also relies on majorizing sequences. The celebrated contraction mapping principle is utilized to study the convergence of the Krasnoselskij-like algorithm. The numerical experimentation demonstrates that the new algorithms are essentially as effective but less expensive to implement. Although the new approach is demonstrated for Newton-like algorithms, it can be applied to other single-step, multistep, or multipoint algorithms using inverses of linear operators along the same lines.

Keywords: hybrid-Newton-like algorithm; fixed sum of operators; Banach space; Fréchet derivative; convergence

MSC: 65G99; 65H10; 49H17; 49M15



check for updates

Citation: Argyros, I.K.; George, S.; Regmi, S.; Argyros, C.I. Hybrid Newton-like Inverse Free Algorithms for Solving Nonlinear Equations. *Algorithms* **2024**, *17*, 154. <https://doi.org/10.3390/a17040154>

Academic Editors: Shuai Li and Dunhui Xiao

Received: 7 March 2024

Revised: 30 March 2024

Accepted: 8 April 2024

Published: 10 April 2024

1. Introduction

Let T_1, T_2 stand for Banach spaces; $\Omega \subset T_1$ be an open and convex set; and $G : \Omega \rightarrow T_2$ denote a differentiable operator according to Fréchet [1–4].

A plethora of applications from diverse areas of research such as Optimization and Computational Sciences are reduced using mathematical modeling [5–14] to locate a solution x^* of a nonlinear equation like

$$G(x) = 0 \quad (1)$$

The closed version of a solution $x^* \in \Omega$ is possible only in special cases. Consequently, most solution approaches utilized by researchers and practitioners are iterative when the sequence is generated approximating the solution x^* .

The algorithm of successive substitutions or the algorithm of iteration or the Picard algorithm is a simple and important algorithm for solving linear as well as nonlinear equations. This algorithm originated in antiquity, appearing in the writings of Heron of Alexandria in the second century B.C. in relation to root extraction. Later, Cauchy, as well as Picard, employed this algorithm to assure the existence of solutions of differential equations. It is defined for $P : T_1 \rightarrow T_1$ and each $n = 0, 1, 2, \dots$ by



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

$$z_0 \in \Omega, z_{n+1} = P(z_n).$$

Banach inaugurated the abstract formulation of this algorithm followed by Cacciopoli and Weisinger. We refer the readers to the reference by Berinde [15], Krasnoselskij [16], and Kantorovich et al. [3] for further information on the convergence conditions (see also [17–23]). Concerning the convergence order of this algorithm, it is only linear. Thus, there is a need for introducing algorithms of convergence order higher than one.

Newton’s algorithm is without a doubt the most well-known algorithm of convergence order two for solving transcendental as well as scalar equations. It is written for each $n = 0, 1, 2, \dots$ as

$$x_0 \in \Omega, x_{n+1} = x_n - G'(x_n)^{-1}G(x_n),$$

where G' is the notation for the derivative of the operator G according to Fréchet [8,24,25]. The construction of Newton’s algorithm is based on linearization. Let $x_0 \in \Omega$ be an initial point. If the operator G is Fréchet-differentiable, one can write the Ostrowski [21] representation

$$G'(x) = G(x_0) + G'(x_0)(x - x_0) + d(x, x_0),$$

where $\|d(x, x_0)\| = O(\|x - x_0\|)$ as $\|x - x_0\| \rightarrow 0$. If $x^* \in \Omega$ is a solution of the Equation (1), it follows by the preceding representation that

$$G(x_0) + G'(x_0)(x^* - x_0) = -d(x^*, x_0).$$

If x^* is near to x_0 , then one can neglect the supposedly small quantity $d(x^*, x_0)$, leading to a linear equation

$$G(x_0) + G'(x_0)(x - x_0) = 0.$$

It is said that this equation is obtained from Equation (1) by the technique of linearization or the tangent algorithm since this is the equation of the tangent line to the curve $y = G(x)$ at the point $(x_0, G(x_0))$ provided that G is a real function. It follows by this linearization that x_1 is a unique solution if $G'(x_0)^{-1}$ exists. In this case, we can write

$$x_1 = x_0 - G'(x_0)^{-1}G(x_0).$$

If x_0 is close to x^* , then maybe x_1 is even closer. Thus, this process can be repeated with x_1 replacing x_0 leading to Newton’s algorithm. A main drawback with the implementation of Newton’s algorithm is the inversion of the linear operator $G'(x_n)$ at each step of the iteration. We address this issue in this paper. Our methodology applies to all single-step, multistep methods, and quasi-Newton [5,10,20,26,27] using inverses along the same lines. We shall demonstrate our methodology to a large class of algorithms involving inverses including Newton’s algorithm as a special case.

Let us consider the popular Newton-like algorithm defined for $y_0 \in \Omega$ and each $n = 0, 1, 2, \dots$ by

$$y_0 \in \Omega, y_{n+1} = y_n - L_n^{-1}G(y_n) \tag{2}$$

Let $L : \Omega \rightarrow \mathcal{L}(T_1, T_2)$, the space of bounded linear operators mapping the space T_1 into T_2 . Notice that for $L_n = G'(y_0)$ or $L_n = G'(y_n)$, we obtain the modified and Newton’s algorithm, respectively. There are two types of convergence usually studied for iterative algorithms: the semi-local and the local analysis. The former uses the condition on the initial guess y_0 , and the operator F and the solution y^* are found in a neighborhood of x_0 . The latter differs from the former since the convergence conditions depend on y^* and demonstrate how difficult is to choose the initial guess y_0 . The main challenge of local analysis is that y^* is usually unknown. Numerous papers have been presented dealing with the semi-local as well as the local analysis of convergence for the Newton-like algorithm (2) [2,24–26,28]. The convergence conditions involve Lipchitz–Holder or generalized continuity conditions utilized to control the Fréchet derivative G' of the operator F . By L_n

in (2), we mean $L_n = L(y_n)$. The inversion of the linear operator at each step is computationally expensive or impossible in general. To essentially utilize the algorithm but without the inverse, we replace it with a finite sum of linear operators related to L as follows:

Suppose:

There exists $\Delta \in \mathcal{L}(T_1, T_2)$ such that $\Delta^{-1} \in \mathcal{L}(T_2, T_1)$ and $(I - \Delta^{-1}(\Delta - L(y)))^{-1} \in \mathcal{L}(T_2, T_1)$. Then, the Newton-like algorithm (2) can be rewritten as

$$y_0 \in \Omega, \\ y_{n+1} = y_n - [I - \Delta^{-1}(\Delta - L_n)]^{-1} \Delta^{-1} G(y_n), \tag{3}$$

where I denotes the identity operator. The iterates of algorithms (2) and (3) coincide, since

$$[I - \Delta^{-1}(\Delta - L_n)]^{-1} \Delta^{-1} \\ = [\Delta(I - \Delta^{-1}(\Delta - L_n))]^{-1} = L_n^{-1}.$$

Even if we replace algorithm (2) with algorithm (3), we still need to invert the linear operator $(I - \Delta^{-1}(\Delta - L_n))$ at each step of the iteration. But we can avoid this inversion if we introduce for k a fixed natural number the operators $\Gamma = \Gamma(x) = \Delta^{-1}(\Delta - L(x))$ and

$$A_k(x) = A = I + \Gamma + \Gamma^2 + \dots + \Gamma^k.$$

Then, consider the replacement of algorithm (2) defined for $x_0 = y_0 \in \Omega$ by

$$x_0 \in \Omega, x_{n+1} = x_n - A\Delta^{-1}G(x_n). \tag{4}$$

Algorithm (4) requires the inversion of only the frozen linear operator Δ at each step. Notice also that A is a linear operator. By letting $k \rightarrow +\infty$, and if $\lim_{k \rightarrow +\infty} A_k = L$ exists, then

$$L_n^{-1} = L\Delta^{-1}.$$

The condition

$$\|\Delta^{-1}(\Delta - L(x))\| < 1 \quad x \in \Omega$$

guarantees the existence of this limit [5,8,16]. A possible choice for $\Delta = I$. If $T_1 = T_2 = \mathbb{R}^i$, i a natural number, and H denotes the Hessian of the operator G , then we can choose $\Delta = H(x_0)$ (semi-local case) or $\Delta = H(x^*)$ (local case). The choice $\Delta = H(\bar{x})$ has been considered in [29], where $\bar{x} \in \Omega$ is an auxiliary point. In the more general setting of Banach space $\Delta \in L(T_1, T_2)$ which is the space of bounded linear operator from T_1 into T_2 . As a further example, if $L_n = G'(x_n)$, then $\Delta = G'(x_n)$ (semi-local case) or $\Delta = G'(x^*)$ (local case) or $\Delta = G'(\bar{x})$. Other choices for Δ are possible as long as they satisfy the convergence conditions (H_4) and (H_5) (semi-local case) and (C_1) and (C_2) (local case) (see Section 2 and Section 3, respectively).

We also study the Krasnoselskij-like or the Picard-like algorithm [15,16]

$$x_0 \in \Omega, x_{n+1} = P_k(x_n), \tag{5}$$

where

$$P_k(x) = P(x) = (1 - \lambda)x - \lambda(A\Delta^{-1}G(x) - x), \lambda \in (0, 1]$$

for locating fixed points. If $\lambda = 1$, then algorithm (5) reduces to algorithm (4). The semi-local analysis of convergence for algorithm (4) relies on majorizing sequences [3,24,25]. But the analysis for algorithm (5) depends on the celebrated contraction mapping principle [3,17,21,22].

The preceding reasoning justifies the study of the semi-local and local analysis of convergence appearing in this paper. The rest of the paper is organized as follows: In Sections 2 and 3, we develop the semi-local and local analysis of convergence for algorithm (4). The convergence of the Krasnoselskij-like algorithm is presented in Section 4.

The numerical experimentations demonstrating the efficiency of the new hybrid algorithms are provided in Section 5. Concluding remarks and directions of future research complete this paper in Section 6.

2. Semi-Local Analysis

Throughout this paper, we use the symbol $U(x, \alpha)$ to denote the open ball centered at $x \in T_1$ with radius $\alpha > 0$, and $U[x, \alpha]$ is the closure of $U(x, \alpha)$. The following Banach Lemma is used to prove our results.

Theorem 1 (Banach Lemma on Invertible Operators [3,15]). *If M is a bounded linear operator in T_1 , M^{-1} exists if and only if there is a bounded linear operator M_1 in T_1 such that M_1^{-1} exists and*

$$\|I - M_1M\| < 1.$$

If M^{-1} exists, then

$$M^{-1} = \sum_{n=0}^{\infty} (I - M_1M)^n M_1$$

and

$$\|M^{-1}\| \leq \frac{\|M_1\|}{1 - \|I - M_1M\|}.$$

Further, we use majorizing sequences to prove the semi-local convergence. Recall the definition of a majorizing sequence.

Definition 1 ([3,5]). *Let $\{x_n\}$ be a sequence in a normed space X . Then a nonnegative scalar sequence $\{v_n\}$ for which*

$$\|x_{n+1} - x_n\| \leq v_{n+1} - v_n \quad \forall n \geq 0 \tag{6}$$

holds is a majorizing sequence for $\{x_n\}$. Note that any majorizing sequence is necessarily nondecreasing. Moreover, if the sequence $\{v_n\}$ converges, then $\{x_n\}$ converges too, and for $v^* = \lim_{n \rightarrow \infty} v_n$

$$\|x^* - x_n\| \leq v^* - v_n.$$

Hence, the study of the convergence of the sequence $\{x_n\}$ reduces to that of $\{v_n\}$.

Let $S = [0, +\infty)$.

Suppose:

(H_1) There exists parameters $\delta \geq 0, \gamma \in [0, \frac{1}{2})$, an element $x_0 \in \Omega$, and an invertible operator Δ such that

$$\|A\Delta^{-1}G(x_0)\| \leq \delta.$$

(H_2) There exists a function $w_0 : S \rightarrow S$, which is nondecreasing as well as continuous (FNDC), such that the equation $w_0(t) - 1 = 0$ has the smallest positive solution. Denote such solution by ρ and set $S_0 = [0, \rho)$.

(H_3) There exist (FNDC) $w : S_0 \rightarrow S$ and $w_1 : S_0 \rightarrow S$. Define the scalar sequence $\{\alpha_n\}$ for $\alpha_0 = 0, \alpha_1 = \delta$ some $\gamma \geq 0, \bar{b} \geq 0$, and each $n = 0, 1, 2, \dots$ by

$$q_{n+1} = (1 + \gamma + \dots + \gamma^k) \left(\int_0^1 w((1 - \tau)(\alpha_{n+1} - \alpha_n)) d\tau + w_1(\alpha_n) + \bar{b}\gamma^{k+1} \right),$$

and

$$\alpha_{n+2} = \alpha_{n+1} + q_{n+1}(\alpha_{n+1} - \alpha_n). \tag{7}$$

The sequence $\{\alpha_n\}$ is shown to be majorizing for algorithm (4) in Theorem 2. But first, the convergence conditions are given for the sequence $\{\alpha_n\}$.

(H4) There exists $\rho_0 \in [0, \rho)$ such that for each $n = 0, 1, 2, \dots$
 $q_{n+1} < 1$ and $\alpha_n \leq \rho_0$.

This condition and the formula (7) imply that for each $n = 0, 1, 2, \dots$

$$0 \leq \alpha_n \leq \alpha_{n+1} \leq \rho_0$$

and there exists $\alpha \in [0, \rho_0)$ such that $\lim_{n \rightarrow +\infty} \alpha_n = \alpha$. The parameter α is the unique least upper bound of the sequence $\{\alpha_n\}$.

It is worth noting that this sequence can be computed a priori and relates to the initial approximation. Such conditions are weaker than the usual convergence conditions given as functions of the starting point [3–5].

Next, we relate the scalar sequences and functions w_0, w to the operators on algorithm (4).

(H5) $\|\Delta^{-1}(G'(v) - \Delta)\| \leq w_0(\|v - x_0\|)$ for each $v \in \Omega$.

Set $\Omega_0 = U(x_0, \rho) \cap \Omega$.

(H6)

$$\|\Delta^{-1}(G'(v_2) - G'(v_1))\| \leq w(\|v_2 - v_1\|), \quad \gamma = \|\Delta^{-1}(\Delta - L(v))\|,$$

and

$$\|\Delta^{-1}(G'(v) - L(v))\| \leq w_1(\|v - x_0\|)$$

for each $v, v_1, v_2 \in \Omega_0$.

Set $b = \gamma \frac{1-\gamma^k}{1-\gamma}$ and $\bar{b} = \frac{1}{1-b}$.

and

(H7) $U[x_0, \alpha] \subset \Omega$.

The conditions (H1)–(H5) and the developed notations are utilized to show the main semi-local analysis of convergence for algorithm (4).

Theorem 2. *Suppose that the conditions (H1)–(H5) hold. Then, if the initial guess $x_0 \in \Omega$ the following assertions hold for the sequence $\{x_n\}$ generated by algorithm (4)*

$$\{x_n\} \subset B(x_0, \alpha) \tag{8}$$

and there exists a solution $x^* \in B[x_0, \alpha]$ of the equation $G(x) = 0$ such that

$$\|x^* - x_n\| \leq \alpha - \alpha_n, \tag{9}$$

where the sequence $\{\alpha_n\}$ is given by the formula (7) and α is its limit.

Proof. Notice that all the iterates $\{x_m\}$ (m a natural number) of algorithm (4) are well defined. We present a proof based on mathematical induction. In particular, we show that for each $n = 0, 1, 2, \dots$

$$\|x_{n+1} - x_n\| \leq \alpha_{n+1} - \alpha_n. \tag{10}$$

Assertion (9) holds if $n = 0$, by (4), (7), and the condition (H1), since

$$\|x_1 - x_0\| = \|A\Delta^{-1}G(x_0)\| \leq \alpha_1 - \alpha_0 < \alpha.$$

It also follows that the iterate $x_1 \in U(x_0, \alpha)$. Next, we show that the linear operator $A = A(x_m)$ is invertible by using the restrictions on γ, b, \bar{b} given in the condition (H_5) :

$$\begin{aligned} \|I - A(x_m)\| &= \|\Gamma + \Gamma^2 + \dots + \Gamma^k\| \\ &\leq \|\Gamma\| + \|\Gamma\|^2 + \dots + \|\Gamma\|^k = \|\Gamma\| \frac{1 - \|\Gamma\|^k}{1 - \|\Gamma\|} \\ &\leq \gamma \frac{1 - \gamma^k}{1 - \gamma} = b < 1 \end{aligned} \tag{11}$$

since $\gamma \in [0, \frac{1}{2})$ by (11) and the condition (H_1) . Inequality (11) and Theorem 1 assure the invertibility of the operator A , and

$$\|A^{-1}\| \leq \frac{1}{1 - b} = \bar{b}. \tag{12}$$

Then, we can write by algorithm (2) in turn that

$$\begin{aligned} G(x_{m+1}) &= G(x_{m+1}) - G(x_m) - \Delta A^{-1}(x_{m+1} - x_m) \\ &= G(x_{m+1}) - G(x_m) - G'(x_m)(x_{m+1} - x_m) \\ &\quad + (G'(x_m) - L_m)(x_{m+1} - x_m) \\ &\quad + (L_m - \Delta A^{-1})(x_{m+1} - x_m) \\ &= G(x_{m+1}) - G(x_m) - G'(x_m)(x_{m+1} - x_m) \\ &\quad + (G'(x_m) - L_m)(x_{m+1} - x_m) \\ &\quad + (L_m A - \Delta) A^{-1}(x_{m+1} - x_m). \end{aligned} \tag{13}$$

But we have

$$\begin{aligned} L_m A - \Delta &= L_m(I + \Gamma + \Gamma^2 + \dots + \Gamma^k) - \Delta \\ &= L_m - \Delta + (-\Delta + \Delta + L_m)(\Gamma + \Gamma^2 + \dots + \Gamma^k) \\ &= L_m - \Delta + \Delta(\Gamma + \Gamma^2 + \dots + \Gamma^k) \\ &\quad - (\Delta - L_m)(\Gamma + \Gamma^2 + \dots + \Gamma^k) \\ &= L_m - \Delta + \Delta\Gamma + \Delta(\Gamma^2 + \dots + \Gamma^k) \\ &\quad - (\Delta - L_m)(\Gamma + \Gamma^2 + \dots + \Gamma^k) \\ &= \Delta(\Gamma^2 + \dots + \Gamma^k) - (\Delta - L_m)(\Gamma + \Gamma^2 + \dots + \Gamma^k) \end{aligned} \tag{14}$$

since $L_m - \Delta + \Delta\Gamma = 0$ by the definition of Γ . Thus, we obtain from (14)

$$\begin{aligned} \Delta^{-1}(L_m A - \Delta) &= \Gamma^2 + \dots + \Gamma^k - \Gamma(\Gamma + \Gamma^2 + \dots + \Gamma^k) \\ &= -\Gamma^{k+1}. \end{aligned} \tag{15}$$

Then, it follows by (13), the conditions (H_4) , (H_5) , (13), the conditions (H_4) , (H_5) , (15), the inductions hypotheses, and the triangle inequality, in turn, that

$$\begin{aligned} \|\Delta^{-1}G(x_{m+1})\| &\leq \int_0^1 w((1 - \tau)\|x_{m+1} - x_m\|)d\tau \|x_{m+1} - x_m\| \\ &\quad + w_1(\|x_m - x_0\|)\|x_{m+1} - x_m\| + \bar{b}\gamma^{k+1}\|x_{m+1} - x_m\| \\ &\leq \left[\int_0^1 w((1 - \tau)(\alpha_{m+1} - \alpha_m)d\tau + w_1(\alpha_m) + \bar{b}\gamma^{k+1} \right] \\ &\quad \times (\alpha_{m+1} - \alpha_m). \end{aligned} \tag{16}$$

Consequently, by algorithms (4), (7), and (16), we obtain in turn that

$$\begin{aligned} \|x_{m+2} - x_{m+1}\| &\leq \|A(x_{m+1})\Delta^{-1}G(x_{m+1})\| \\ &\leq \|A(x_{m+1})\| \|\Delta^{-1}G(x_{m+1})\| \\ &\leq (1 + \gamma + \dots + \gamma^k) \left[\int_0^1 w((1 - \tau)(\alpha_{m+1} - \alpha_m)d\tau \right. \\ &\quad \left. + w_1(\alpha_m) + \bar{b}\gamma^{k+1} \right] (\alpha_{m+1} - \alpha_m) = \alpha_{m+2} - \alpha_{m+1}, \end{aligned}$$

and

$$\begin{aligned} \|x_{m+2} - x_0\| &\leq \|x_{m+2} - x_{m+1}\| + \|x_{m+1} - x_0\| \\ &\leq \alpha_{m+2} - \alpha_{m+1} + \alpha_{m+1} - \alpha_0 = \alpha_{m+2} < \alpha. \end{aligned}$$

Hence, assertion (10) holds and the iterate $x_{m+2} \in B(x_0, \alpha)$. By the condition (H_4) , the sequence $\{\alpha_m\}$ is complete as convergent to α . Therefore, by (10) the sequence $\{x_m\}$ is also complete in the Banach space T_1 , and as such it converges to some $x^* \in B[x_0, \alpha]$ (since $B[x_0, \alpha]$ is a closed set). By sending $m \rightarrow +\infty$ in (16), and the continuity of the operator G , we deduce that $G(x^*) = 0$. Finally, the estimation for i a natural number

$$\|x_{m+i} - x_m\| \leq \alpha_{m+i} - \alpha_m,$$

shows (9), if $i \rightarrow +\infty$. \square

Next, a set is specified that contains only one solution of the equation $G(x) = 0$.

Proposition 1. *Suppose: There exists a solution $y \in B(x_0, \rho_1)$ of the equation $G(x) = 0$ for some $\rho_1 > 0$; the condition (H_4) holds in the ball $B(x_0, \rho_1)$, and there exists $\rho_2 \geq \rho_1$ such that*

$$\int_0^1 w_0(\tau\rho_1 + (1 - \tau)\rho_2)d\tau < 1 \tag{17}$$

Set $\Omega_2 = B[x_0, \rho_2] \cap \Omega$. Then, the element y is the only solution of the equation $G(x) = 0$ in the set Ω_2 .

Proof. Suppose that there exists a solution $z \in \Omega_2$ of the equation $G(x) = 0$ with $z \neq y$. Define the linear operator $E = \int_0^1 G'(y + \tau(z - y))d\tau$. By using the condition (H_4) and (17), we obtain in turn

$$\begin{aligned} \|\Delta^{-1}(E - \Delta)\| &\leq \int_0^1 w_0(\tau\|y - x_0\| + (1 - \tau)\|z - x_0\|)d\tau \\ &\leq \int_0^1 w_0(\tau\rho_2 + (1 - \tau)\rho_3)d\tau < 1. \end{aligned}$$

Thus, the linear operator $E^{-1} \in \mathcal{L}(T_2, T_1)$, and from the identity we obtain

$$z - y = E^{-1}(G(z) - G(y)) = E^{-1}(0) = 0.$$

Therefore, we conclude that $z = y$. \square

Remark 1.

- (1) The limit point α can be replaced by ρ in the condition (H_6) .
- (2) If all the conditions (H_1) - (H_6) hold in Proposition 1, take $y = x^*$ and $\rho_2 = \alpha$.
- (3) The second hypothesis in the condition (H_5) can be replaced as follows:
 Suppose that there exists (FNDC) $w_2 : S \rightarrow S$ such that equation $w_2(t) - 1 = 0$ has an SPS. Denote such solution by ρ_3 , and set $\gamma \in [0, w_2(\rho_3))$. Then, $\gamma \in [0, 1)$. In this case, set $\bar{\alpha} = \min\{\alpha, \rho_3\}$, and replace α by $\bar{\alpha}$ in the condition (H_6) .

- (4) The results for algorithm (3) or (2) can be obtained if we let $k \rightarrow +\infty$ in Theorem 1. A possible choice for $k = n$, although a smaller value $k = 1$ or $k = 2$ is preferred to reduce the computational cost (see also the numerical Section 6).

3. Local Analysis

In Section 3, we exchange the role of x^* and the “ w ” functions with x_0 , and the “ ψ ” function, respectively. But the computations are similar.

Suppose:

- (C₁) There exists a solution $x^* \in \Omega$ of the equation $G(x) = 0$, and an invertible operator $\Delta \in \mathcal{L}(T_1, T_2)$ such that for each $\tau \in [0, 1]$, $x \in \Omega$

$$\begin{aligned} & \left\| \int_0^1 \Delta^{-1}(G'(x^* + \tau(x - x^*)) - \Delta) d\tau \right\| \\ & \leq \int_0^1 \psi(\tau \|x - x^*\|) d\tau, \end{aligned}$$

for some (FNDC) $\psi : S \rightarrow S$.

- (C₂) There exists $\gamma \in [0, \frac{1}{2})$, such that

$$\|\Delta^{-1}(A(x) - \Delta)\| \leq \gamma.$$

Define the function $g : S \rightarrow S$ by

$$g(t) = \frac{1 - \gamma^{k+1}}{1 - \gamma} \int_0^1 \psi(\tau t) d\tau + \gamma \frac{1 - \gamma^k}{1 - \gamma}.$$

- (C₃) The equation $g(t) - 1 = 0$ has an SPS. Denote such a solution by ρ_4 .
and
(C₄) $U[x^*, \rho_4] \subset \Omega$.

Theorem 3. Suppose that the conditions (C₁)–(C₄) hold. Then, the sequence $\{x_n\}$ with initial guess $x_0 \in U(x^*, \rho_4) - \{x^*\}$ exists in $U(x^*, \rho_4)$, stays in $U(x^*, \rho_4)$, and converges to x^* such that

$$\|x_{n+1} - x^*\| \leq g(\|x_n - x^*\|) \|x_n - x^*\| \leq \|x_n - x^*\| \leq \rho_4. \tag{18}$$

Proof. Assertion (18) is shown by mathematical induction. Notice that all the iterates x_m exist by algorithm (4). We can also write, in turn, that

$$\begin{aligned} x_{m+1} - x^* &= x_m - x^* - A\Delta^{-1}G(x_m) \\ &= x_m - x^* - A\Delta^{-1} \int_0^1 G'(x^* + \tau(x_m - x^*)) d\tau (x_m - x^*) \\ &= [I - A\Delta^{-1}(\int_0^1 G'(x^* + \tau(x_m - x^*)) - \Delta + \Delta) d\tau](x_m - x^*) \\ &= [(I - A - A\Delta^{-1}(\int_0^1 G'(x^* + \tau(x_m - x^*)) - \Delta) d\tau)](x_m - x^*). \end{aligned} \tag{19}$$

But, we have as in the semi-local case

$$\|A\| \leq \frac{1 - \gamma^{k+1}}{1 - \gamma},$$

and

$$\|I - A\| \leq \gamma \frac{1 - \gamma^k}{1 - \gamma}. \tag{20}$$

Hence, (19) turns into

$$x_{m+1} - x^* = \left[(I - A) - A \left(\int_0^1 \Delta^{-1}(G'(x^* + \tau(x_m - x^*))) - \Delta \right) d\tau \right] (x_m - x^*). \tag{21}$$

Using the conditions (C₁)–(C₃), we obtain

$$\begin{aligned} \|x_{m+1} - x^*\| &\leq \left[\gamma \frac{1 - \gamma^k}{1 - \gamma} + \frac{1 - \gamma^{k+1}}{1 - \gamma} \int_0^1 \psi(\tau \|x_m - x^*\|) d\tau \right] \|x_m - x^*\| \\ &\leq g(\|x_m - x^*\|) \|x_m - x^*\| \leq \|x_m - x^*\| < \rho_4. \end{aligned} \tag{22}$$

Thus, assertion (18) holds and the iterate $x_{m+1} \in U(x^*, \rho_4)$. Then, by (22), we obtain

$$\|x_{m+1} - x^*\| \leq d \|x_m - x^*\| \leq d^{m+1} \|x_0 - x^*\|, \tag{23}$$

where $d = g(\|x_0 - x^*\|) \in [0, 1)$. Therefore, we deduce from (23) that $\lim_{m \rightarrow +\infty} x_m = x^*$, and the iterate $x_{m+1} \in U(x^*, \rho_4)$. □

Next, the uniqueness of the x^* set is determined.

Proposition 2. *Suppose: There exists $\rho_5 > 0$ such that the condition (C₁) holds in the ball $U(x^*, \rho_5)$,*

$$\|\Delta^{-1}(G'(u) - M)\| \leq \psi_0(\|u - x^*\|) \tag{24}$$

for each $u \in \Omega$ and some (FNDC) $\psi_0 : S \rightarrow S$ and there exists $\rho_6 \geq \rho_5$ such that

$$\int_0^1 \psi_0(\tau \rho_6) d\tau < 1. \tag{25}$$

Define the set $\Omega_2 = U[x^*, \rho_6] \cap \Omega$. Then, the only solution of the equation $G(x) = 0$ in the set Ω_2 is x^* .

Proof. Suppose that there exists $y \in \Omega_3$ solving the equation $G(x) = 0$ and $y_0 \neq x^*$. Define the linear operator $E_1 = \int_0^1 G'(x^* + \tau(y_0 - x^*)) d\tau$. By the condition (C₁), (24), and (25), we have in turn that

$$\begin{aligned} \|\Delta^{-1}(E_1 - \Delta)\| &\leq \int_0^1 \psi_0(\tau \|y_0 - x^*\|) d\tau \\ &\leq \int_0^1 \psi_0(\tau \rho_6) d\tau < 1, \end{aligned}$$

Hence, L_1 is invertible. Finally, from the identity $y_0 - x^* = E^{-1}(G(y_0) - G(x^*)) = E^{-1}(0) = 0$, we conclude that $y_0 = x^*$. □

Remark 2.

- (1) We can set in Proposition 2 $y_0 = x^*$ and $\rho_5 = \rho_4$.
- (2) The parameter ρ_4 defined in the condition (C₃) is the radius of convergence for algorithm (4).
- (3) As in the semi-local analysis if $k \rightarrow +\infty$, we obtain the results for algorithm (3). Another choice for $k = n$. However, we shall choose a small value of k to save computational cost.

4. Convergence of the Krasnoselskij-like Algorithm

The contraction mapping principle has been used extensively to find fixed points using iterative algorithms.

Theorem 4 ([15]). Let $Q : T_1 \rightarrow T_1$ be a contraction operator, with Lipchitz parameter $\zeta \in [0, 1)$. Then, the operator Q has a fixed point $x^* \in T_1$ i.e., $Q(x^*) = x^*$. Moreover, for each initial guess $x_0 \in T_1$ the Picard algorithm or the algorithm of successive substitutions $x_{n+1} = Q(x_n)$ converges to $x^* \in T_1$.

Theorem 4 cannot be used if the operator Q has more than one fixed point. The fixed points must be separated in this case. Let us consider Q_0 to be a closed substep of Q_0 with $Q_0 \neq \phi$. Then, the following result is available.

Theorem 5 ([15]). Suppose that the operator $Q : Q_0 \rightarrow Q_0$ is a contraction with constant $\zeta \in [0, 1)$. Then, the operator Q has a unique fixed point $x^* \in Q_0$. Moreover, for each $x_0 \in Q_0$, the Picard algorithm converges to x^* . The convergence of algorithm (5) is based on Theorem 6.

Theorem 6. Let k be a fixed natural number. Suppose that the following conditions hold for $x, y \in \Omega$:

$$\begin{aligned} \|\Delta^{-1}(L(x) - \Delta)\| &< 1, \\ \|P(x) - P(x_0)\| &\leq p_0\|x - x_0\|, \\ \|P(y) - P(x_0)\| &\leq p\|y - x\| \end{aligned}$$

and

$$\|P(x_0) - x_0\| \leq \frac{r}{1 - p_0},$$

for some invertible operator and $p_0, p \in [0, 1)$, and $U(x_0, r) \subset \Omega$ provided that $r \in [0, \frac{1}{p_0})$. Then, the operator $P : \Omega \rightarrow \Omega$ has a unique fixed point $x^* \in \Omega$, and the Krasnoselskij-like algorithm (5) converges to x^* .

Proof. Notice that

$$\begin{aligned} \|P(x) - x_0\| &\leq \|P(x) - P(x_0)\| + \|P(x_0) - x_0\| \\ &\leq p_0r + \|P(x_0) - x_0\| \leq r. \end{aligned}$$

Thus, $P : U(x_0, r) \rightarrow U(x_0, r)$ is a contraction operator with constant $p \in [0, 1)$. Set $Q_0 = U[x_0, r]$. Then, the result follows from Theorem 5. \square

5. Error Analysis

The sequences $\{y_n\}$ and $\{x_n\}$ are generated by formulas (2) and (4), respectively. We select a portion of the standard semi-local convergence result for the Newton-like algorithm (2) [28].

Theorem 7. Let $G : \Omega \rightarrow Y$ be Fréchet-differentiable and let $L(x) \in L(T_1, T_2)$ be an approximation to the linear operator $G'(x)$. Suppose that there exist an open convex subset $\tilde{\Omega}$ of Ω , $x_0 \in \tilde{\Omega}$, a bounded linear invertible operator $K_0(= L(x_0))$, and constants $\eta, K > 0, K_0, K_1, \mu, l \geq 0$ such that for all $x, y \in \tilde{\Omega}$ the following conditions hold:

$$\begin{aligned} \|L_0^{-1}G(x_0)\| &\leq \eta, \\ \|K_0^{-1}(F'(y) - F'(x))\| &\leq K\|y - x\|, \\ \|K_0(F'(x) - K_1(x))\| &\leq K_0\|x - x_0\| + \mu, \\ \|K_0(A(x) - K_0)\| &\leq K_1\|x - x_0\| + l, \\ l_1 &= \mu + l < 1. \end{aligned}$$

In addition, suppose that

$$h := \sigma\eta \leq \frac{1}{2}(1 - l_1)^2,$$

where $\sigma = \max\{K, K_0 + K_1\}$, and

$$\bar{U} = U[x_0, t^*] \subset \tilde{\Omega}$$

where

$$t^* = (1 - l_1 - \sqrt{(1 - l_1)^2 - 2h}) / \sigma_0$$

Then, the following assertions hold

The sequence $\{y_n\}$ generated by algorithm (2) remains in the ball $U(x_0, t^*)$ and converges to a solution $x^* \in \bar{U}$ of the equation $G(x) = 0$ and

$$\|L_n^{-1}K_0\| \leq s_1 := \frac{1}{1 - (K_0t^* + l)} \tag{26}$$

Next, the sequences $\{y_n\}$ and $\{x_n\}$ are related to each other.

Lemma 1. Suppose that the conditions $(H_1) - (H_5)$ for $(\Delta = K_0)$, and those of Theorem 7 hold. Then, the following error assertion holds for each $n = 0, 1, 2, \dots$

$$\|x_{n+1} - y_{n+1}\| \leq e_n, \tag{27}$$

where

$$e_n = s_1 \left[\left(\int_0^1 w((1 - \tau)\|x_n - y_n\|) d\tau + w(\|x_n - y_n\|) + w_1(\|x_n - x_0\|) \|x_n - y_n\| \right) + \bar{b}\gamma^{k+1} \|x_{n+1} - x_n\| \right]. \tag{28}$$

Proof. Under the conditions of Theorems 2 and 7, the iterates $\{y_n\}$ and $\{x_n\}$ are well defined by formulas (2), and (4), respectively. By subtracting (2) from (4) and pulling out L_n^{-1} , we can write, in turn, that

$$\begin{aligned} x_{n+1} - y_{n+1} &= x_n - y_n + L_n^{-1}G(y_n) - \Delta\Delta^{-1}G(x) \\ &= L_n^{-1}[L_n(x_n - y_n) - G(x_n) + G(x_n) + G(y_n) - L_n\Delta\Delta^{-1}G(x_n)] \\ &= -L_n^{-1}(G(x_n) - G(y_n) - F'(y_n)(x_n - y_n)) \\ &\quad - L_n^{-1}(F'(y_n) - F'(x_n))(x_n - y_n) - L_n^{-1}(F'(x_n) - L_n)(x_n - y_n) \\ &\quad + L_n^{-1}(\Delta - L_nA)A^{-1}(x_{n+1} - x_n). \end{aligned} \tag{29}$$

We need, in turn, the following estimates obtained by the conditions of Theorems 2 and 7

$$\begin{aligned} &\|K_0^{-1}(G(x_n) - G(y_n) - F'(y_n)(x_n - y_n))\| \\ &\leq \left\| \int_0^1 sK_0^{-1}(F'(y_n + \tau(x_n - y_n)) - F'(y_n))d\tau(x_n - y_n) \right\| \\ &\leq \int_0^1 sw((1 - \tau)\|x_n - y_n\|)d\tau \|x_n - y_n\| \end{aligned} \tag{30}$$

$$\|K_0^{-1}(F'(y_n) - F'(x_n))(x_n - y_n)\| \leq w(\|x_n - y_n\|) \|x_n - y_n\|, \tag{31}$$

$$\begin{aligned} \|K_0^{-1}(F'x_n - L_n)(x_n - y_n)\| &\leq w_1(\|x_n - x_0\|) \|x_n - y_n\| \\ &\leq w_1(\alpha) \|x_n - y_n\| \end{aligned} \tag{32}$$

$$\|\Delta^{-1}(\Delta - L_nA)\| \Gamma^{k+1} \leq \gamma^{k+1}, \tag{33}$$

and

$$\|K_0G(x_n)\| = \|B^{-1}(x_{n+1} - x_n)\| = \bar{b}\|x_{n+1} - x_n\| \tag{34}$$

By summing up (30)–(34) and using the triangle inequality in (34), we obtain (27). \square

It is convenient for the next result to define the function $\varphi : [0, +\infty) \rightarrow R$ by

$$\varphi(t) = s_1 \left[\left(\int_0^1 w((1-\tau)t) d\tau + w(t) + w_1(\alpha) \right) t + s_2 \right] - t, \text{ where } s_2 = s_1 \bar{b} \gamma^{k+1} \eta.$$

Proposition 3. *Let all the conditions of Lemma 1 hold. Suppose, in addition, that the equation $\psi(t) = 0$ has the smallest solution $s^* > s_2$. Then, the following assertion holds for each $n = 0, 1, 2, \dots$*

$$\|x_n - y_n\| \leq s^*. \tag{35}$$

Proof. The estimate (27) for $n = 0$ implies $\|x_1 - y_1\| \leq s_2 \leq s^*$, which is true by the choice of s^* . Suppose that (35) holds for all integer values m smaller or equal to m . Then, we have by the induction hypothesis, and (2) that

$$\|x_{m+1} - y_{m+1}\| \leq e_m \leq s_1 \left(\int_0^1 w((1-\tau)s^*) d\tau + w(s^*) + w_1(\alpha) \right) s^* + s_2 = s^*, \tag{36}$$

by the definition of s^* . \square

6. Numerical Examples

The examples use $L_n = G'(x_n)$, $\Delta = I$, which are independent of x_0 and x^* .

Example 1. *The solution sought for the nonlinear system*

$$\begin{aligned} g_1 &= u - 0.1 \sin u - 0.3 \cos v + 0.4 \\ g_2 &= v - 0.2 \cos u + 0.1 \sin v + 0.3. \end{aligned}$$

Let $G = (g_1, g_2)$. Then, the system becomes

$$G(s) = 0 \text{ for } s = (\theta_1, \theta_2)^T.$$

Then

$$G'((u, v)) = \begin{bmatrix} 1 - 0.1 \cos(u) & 0.3 \sin(v) \\ 0.2 \sin(u) & 0.1 \cos(v) + 1 \end{bmatrix}.$$

Algorithm (2)

$$x_{k+1} = x_k - G'(x_k)^{-1} G(x_k).$$

Algorithm (4), $k = 1$, $\Delta = I$

$$\begin{aligned} A_1(x) &= I + (I - G'(x)), \\ p_1(x) &= x - (I + (I - G'(x)))G(x), \\ x_{n+1} &= p_1(x_n). \end{aligned} \tag{37}$$

Algorithm (4), $k = 2$, $\Delta = I$

$$\begin{aligned} A_2(x) &= I + (I - G'(x)) + (I - G'(x))^2, \\ p_2(x) &= x - A_2(x)G(x), \\ x_{n+1} &= p_2(x_n). \end{aligned} \tag{38}$$

Algorithm (4), $k = 3$, $\Delta = I$

$$\begin{aligned} A_3(x) &= I + (I - G'(x)) + (I - G'(x))^2 + (I - G'(x))^3, \\ p_3(x) &= x - A_3(x)G(x), \\ x_{n+1} &= p_3(x_n). \end{aligned} \tag{39}$$

Algorithm (4), $k = 4, \Delta = I$

$$\begin{aligned}
 A_4(x) &= I + (I - G'(x)) + (I - G'(x))^2 + (I - G'(x))^3 + (I - G'(x))^4, \\
 p_4(x) &= x - A_4(x)G(x), \\
 x_{n+1} &= p_4(x_n).
 \end{aligned}
 \tag{40}$$

Algorithm (4), $k = 5, \Delta = I$

$$\begin{aligned}
 A_5(x) &= I + (I - G'(x)) + (I - G'(x))^2 + (I - G'(x))^3 + (I - G'(x))^4 + (I - G'(x))^5, \\
 p_5(x) &= x - A_5(x)G(x), \\
 x_{n+1} &= p_5(x_n).
 \end{aligned}
 \tag{41}$$

Algorithm (4), $k = \overline{1,5}, \Delta = G'(x_0)$

$$\begin{aligned}
 x_{n+1} &= x_n - A\Delta^{-1}G(x_n), \\
 \Gamma &= \Delta^{-1}(\Delta - G'(x)), \\
 A &= I + \sum_{i=1}^k \Gamma^i.
 \end{aligned}
 \tag{42}$$

Thus, the comparison shows that the behavior of method (4) is essentially the same as Newton’s method (2). However, the iterates of method (4) are cheaper to obtain than Newton’s. As observed in Tables 1–4, the number of iterations required for the proposed methods with k ranging from 3 to 5 closely aligns with those of Newton’s method.

Table 5 shows the results of calculations to determine the Computational Order of Convergence (COC) and the Approximated Computational Order of Convergence (ACOC) aiming to compare the convergence order of method (4) with the convergence order of Newton’s method (2).

Table 1. Number of iterations to achieve tolerance $\varepsilon = 10^{-9}$ with initial guess $x_0 = (1, 1)$ and $\|I - G'(x_0)\| = 0.3129 < 1$.

Algorithm	Iterations	Algorithm	Iterations
(2) Newton	4	(2) Newton	4
(37), $k = 1$	6	(42), $k = 1$	8
(38), $k = 2$	5	(42), $k = 2$	6
(39), $k = 3$	4	(42), $k = 3$	5
(40), $k = 4$	4	(42), $k = 4$	5
(41), $k = 5$	4	(42), $k = 5$	4

Table 2. Number of iterations to achieve tolerance $\varepsilon = 10^{-9}$ with initial guess $x_0 = (0, 0)$ and $\|I - G'(x_0)\| = 0.1414 < 1$.

Algorithm	Iterations	Algorithm	Iterations
(2) Newton	3	(2) Newton	3
(37), $k = 1$	5	(42), $k = 1$	3
(38), $k = 2$	4	(42), $k = 2$	3
(39), $k = 3$	3	(42), $k = 3$	3
(40), $k = 4$	3	(42), $k = 4$	3
(41), $k = 5$	3	(42), $k = 5$	3

Table 3. Number of iterations to achieve tolerance $\varepsilon = 10^{-9}$, where $x_0 = (-15, -15)$ and $\|I - G'(x_0)\| = 0.257 < 1$.

Algorithm	Iterations	Algorithm	Iterations
(2) Newton	5	(2) Newton	5
(37), $k = 1$	7	(42), $k = 1$	9
(38), $k = 2$	5	(42), $k = 2$	7
(39), $k = 3$	5	(42), $k = 3$	6
(40), $k = 4$	5	(42), $k = 4$	6
(41), $k = 5$	5	(42), $k = 5$	5

Table 4. Number of iterations to achieve tolerance $\varepsilon = 10^{-12}$, where $x_0 = (-15, -15)$ and $\|I - G'(x_0)\| = 0.257 < 1$.

Algorithm	Iterations	Algorithm	Iterations
(2) Newton	7	(2) Newton	7
(37), $k = 1$	8	(42), $k = 1$	12
(38), $k = 2$	7	(42), $k = 2$	8
(39), $k = 3$	7	(42), $k = 3$	8
(40), $k = 4$	7	(42), $k = 4$	7
(41), $k = 5$	7	(42), $k = 5$	7

Definition 2. The computational order of convergence of a sequence $\{x_n\}_{n \geq 0}$ is defined by

$$\bar{\rho}_n = \frac{\ln |e_{n+1}/e_n|}{\ln |e_n/e_{n-1}|},$$

where x_{n-1}, x_n, x_{n+1} are three consecutive iterations near the root α and $e_n = x_n - \alpha$ [6].

Definition 3. The approximated computational order of convergence of a sequence $\{x_n\}_{n \geq 0}$ is defined by

$$\hat{\rho}_n = \frac{\ln |\hat{e}_{n+1}/\hat{e}_n|}{\ln |\hat{e}_n/\hat{e}_{n-1}|},$$

where $\hat{e}_n = x_n - x_{n-1}$. x_n, x_{n-1}, x_{n-2} are three consecutive iterates [6].

Table 5. Computational Order of Convergence and the Approximated Computational Order of Convergence, where $x_0 = (-15, -15)$, $\varepsilon = 10^{-12}$.

Algorithm	COC	ACOC
(2) Newton	1.8624	1.9697
(37), $k = 1$	0.863	1
(38), $k = 2$	0.2695	1.0438
(39), $k = 3$	1.9714	2.3569
(40), $k = 4$	1.8354	1.9453
(41), $k = 5$	1.8642	1.9661
(42), $k = 1$	0.9065	1.0118
(42), $k = 2$	0.5912	0.999
(42), $k = 3$	0.7321	0.9926
(42), $k = 4$	1.933	2.0151
(42), $k = 5$	1.8679	1.9578

Table 5 demonstrates that the convergence of the proposed methods closely corresponds with the convergence of Newton’s method, particularly for values of k ranging from 4 to 5 with the convergence order closely approximating 2.

Example 2. Let $T_1 = T_2 = \mathbb{R}^3$ and $\Omega = U[x^*, 1]$. The mapping G is defined on Ω for $\Theta = (\Theta_1, \Theta_2, \Theta_3)^{tr} \in \mathbb{R}^3$ as

$$G(\Theta) = (\Theta_1, e^{\Theta_2} - 1, \frac{e - 1}{2}\Theta_3^2 + \Theta_3)^{tr}.$$

Then, the definition of the derivative according to Fréchet [2,3,8,13,30] gives for the mapping G that

$$G'(\Theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{\Theta_2} & 0 \\ 0 & 0 & (e - 1)\Theta_3 + 1 \end{bmatrix}.$$

The point $x^* = (0, 0, 0)^{tr}$ solves the equation $G(a) = 0$. Moreover, $G'(x^*) = I$. The conditions of Theorem 3 hold, provided that $\gamma = \gamma(t) = (e - 1)t$, $\psi(t) = (e - 1)t$, and $k = 1$. Then, we can take $\rho_4 \in (0, 0.2909883534)$.

Example 3. Let $U[0, 1]$ stand the space of continuous functions mapping the interval $[0, 1]$ into the real number system. Let $T_1 = T_2 = K[0, 1]$ and $\Omega = U[x^*, 1]$ with $x^*(\kappa) = 0$. The operator G is defined on $U[0, 1]$ as

$$G(z)(\kappa) = z(\kappa) - 4 \int_0^1 \kappa z(\tau)^3 d\tau.$$

Then, the definition of the derivative according to Fréchet [2,3,8,13,30] gives for the operator G

$$G'(z(w))(\kappa) = w(\kappa) - 12 \int_0^1 \kappa \tau z(\tau)^2 w(\tau) d\tau$$

for each $w \in U[0, 1]$. Therefore, the conditions of Theorem 3 hold for $x^* = 0$, $k = 1$ $G'(x^*(\kappa)) = I$ if we choose $\gamma = \gamma(t) = 6t$ and $\psi(t) = 6t$. Then, we obtain $\rho_4 \in (0, 0.8\bar{3})$.

7. Concluding Remarks

The difficulty of implementing the Newton-like algorithms is addressed in this paper. In particular, the computation of L_n^{-1} required at each step of the Newton-like algorithms is avoided with the introduction of algorithm (4) (or algorithm (5)), where the inversion only once of a fixed linear operator is required to implement it. The inverse of the linear operator is exchanged with a finite sum of linear operators related to G' . Both the local and the semi-local convergence analysis of these algorithms is comparable to Newton's in the sense that the number of iteration steps to reach a predetermined tolerance of error is essentially the same. The numerical examples are used to demonstrate that algorithm (4) or algorithm (5) are reliable replacements of the Newton-like algorithms for all practical purposes. We plan to study extensions of the presented algorithms like

$$x_{n+1} = x_n - \tilde{G}(x_n)G(x_n),$$

where \tilde{G} is a conscious approximation to the inverse of a linear operator (like L_n) which may be a divided difference or some other operator [18,30–35].

Author Contributions: Conceptualization, I.K.A., S.G., S.R. and C.I.A.; Algorithm, I.K.A., S.G., S.R. and C.I.A.; methodology, I.K.A., S.G., S.R. and C.I.A.; software, I.K.A., S.G., S.R. and C.I.A.; validation, I.K.A., S.G., S.R. and C.I.A.; formal analysis, I.K.A., S.G., S.R. and C.I.A.; investigation, I.K.A., S.G., S.R. and C.I.A.; resources, I.K.A., S.G., S.R. and C.I.A.; data curation, I.K.A., S.G., S.R. and C.I.A.; writing—original draft preparation, I.K.A., S.G., S.R. and C.I.A.; writing—review and editing, I.K.A., S.G., S.R. and C.I.A.; visualization, I.K.A., S.G., S.R. and C.I.A.; supervision, I.K.A., S.G., S.R. and C.I.A.; project administration, I.K.A., S.G., S.R. and C.I.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: We would like to thank graduate Mykhailo Havdiak from the Department of Optimal Processes, Ivan Franko National University of Lviv, Lviv, Ukraine, for providing Example 1 of this paper.

Conflicts of Interest: The authors declare that there are no conflicts of interest.

References

1. Driscoll, T.A.; Braun, R.J. *Fundamentals of Numerical Computation: Julia Edition*; SIAM: Philadelphia, PA, USA, 2022.
2. Ezquerro, J.A.; Gutierrez, J.M.; Hernandez, M.A.; Romero, N.; Rubio, M.J. The Newton algorithm: From Newton to Kantorovich. *Gac. R. Soc. Mat. Esp.* **2010**, *13*, 53–76. (In Spanish)
3. Kantorovich, L.V.; Akilov, G. *Functional Analysis in Normed Spaces*; Fizmatgiz: Moscow, Russia, 1959; German translation, Akademie-Verlag: Berlin, Germany, 1964; English translation (2nd edition), Pergamon Press: London, UK, 1981, 1964.
4. Proinov, P.D. New general convergence theory for iterative processes and its applications to Newton–Kantorovich type theorems. *J. Complex.* **2010**, *25*, 3–42. [\[CrossRef\]](#)
5. Argyros, I.K. *The Theory and Applications of Iteration Methods with Applications*, 2nd ed.; Engineering Series; CRC Press: Boca Raton, FL, USA; Taylor and Francis Publ.: Abingdon, UK, 2022.
6. Candelario, G.; Cordero, A.; Torregrosa, J.R.; Vassileva, M.P. Generalized conformable fractional Newton-type method for solving nonlinear systems. *Numer. Algor.* **2023**, *93*, 1171–1208. [\[CrossRef\]](#)
7. Darve, E.; Wooters, M. *Numerical Linear Algebra with Julia*; SIAM: Philadelphia, PA, USA, 2021.
8. Deufhard, P. *Newton Algorithms for Nonlinear Problems. Affine Invariance and Adaptive Algorithms*; Springer Series in Computational Mathematics; Springer: Berlin/Heidelberg, Germany, 2004; Volume 35.
9. Kelley, C.T. *Solving Nonlinear Equations with Iterative Methods*; Solvers and Examples in Julia, Fundamentals of Algorithms; SIAM: Philadelphia, PA, USA, 2023.
10. Pollock, S.; Reibold, L.G.; Xiao, M. Anderson-accelerated convergence of Picard iterations for incompressible Navier–Stokes equations. *SIAM J. Numer. Anal.* **2019**, *57*, 615–637. [\[CrossRef\]](#)
11. Rackauckas, C.; Mishra, P.; Gowda, S.; Huang, L. *Sparse Diff Tools.jl*, Julia Package, 2020. Available online: <https://github.com/JuliaDiff/SparseDiffTools.jl> (accessed on 20 February 2024).
12. Rackauckas, C.; Nie, Q. DifferentialEquations.jl—a performant and feature-rich ecosystem for solving differential equations in Julia. *J. Open Res. Softw.* **2017**, *5*, 15. [\[CrossRef\]](#)
13. Revels, J. *Reverse Diff.jl*, Julia Package. 2020. Available online: <https://github.com/JuliaDiff/ReverseDiff.jl> (accessed on 20 February 2024).
14. Uecker, H. *Numerical Continuation and Bifurcation in Nonlinear PDEs*; SIAM: Philadelphia, PA, USA, 2021.
15. Berinde, V. *Iterative Approximation of Fixed Points*; Springer: New York, NY, USA, 2007.
16. Krasnoselskij, M.A. Two remarks on the algorithm of successive approximations. *Uspehi Mat. Nauk.* **1995**, *10*, 123–127. (In Russian)
17. Bian, W.; Chen, X.; Kelley, C.T. Anderson acceleration for a class of nonsmooth fixed-point problems. *SIAM J. Sci. Comp.* **2021**, *43*, S1–S20. [\[CrossRef\]](#)
18. De Dierck, H.; He, Y. Linear asymptotic convergence of Anderson acceleration: Fixed point analysis. *arXiv* **2021**, arXiv:14716v1.
19. Evans, C.; Pollock, S.; Reibold, L.G.; Xiao, M. A proof that Anderson acceleration improves the convergence rate in linearly converging fixed-point methods (but not in those converging quadratically). *SIAM J. Numer. Anal.* **2020**, *58*, 788–810. [\[CrossRef\]](#)
20. Fang, H.R.; Saad, Y. Two classes of multisection methods for nonlinear acceleration. *Numer. Linear Algebra Appl.* **2009**, *16*, 197–221. [\[CrossRef\]](#)
21. Pollock, S.; Reibold, L.G. Anderson acceleration for contractive and noncontractive operators. *IMA Numer. Anal.* **2021**, *41*, 2841–2872. [\[CrossRef\]](#)
22. Padcharoen, A.; Kumam, P.; Chaipunya, P.; Shehu, Y. Convergence of inertial modified Krasnoselskii–Mann iteration with application to image recovery. *Thai J. Math.* **2020**, *18*, 126–142.
23. Zhang, J.; O’Donoghue, B.; Boyd, S. Globally convergent type-I Anderson acceleration for nonsmooth fixed-point iterations. *SIAM J. Optim.* **2020**, *30*, 3170–3197. [\[CrossRef\]](#)
24. Argyros, I.K.; George, S. On the complexity of extending the convergence region for Traub’s algorithm. *J. Complex.* **2020**, *56*, 101423. [\[CrossRef\]](#)
25. Argyros, I.K.; George, S. On a unified convergence analysis for Newton-type algorithms solving generalized equations with the Aubin property. *J. Complex.* **2024**, *81*, 101817. [\[CrossRef\]](#)
26. Catinas, E. The inexact, inexact perturbed, and quasi-Newton algorithms are equivalent models. *Math. Comp.* **2005**, *74*, 291–301. [\[CrossRef\]](#)
27. Erfanifar, R.; Hajariah, M.A. A new multi-step method for solving nonlinear systems with high efficiency indices. *Numer. Algorithms* **2024**. [\[CrossRef\]](#)
28. Yamamoto, T. A convergence theorem for Newton-like algorithms in Banach spaces. *Numer. Math.* **1987**, *51*, 545–557. [\[CrossRef\]](#)
29. Ezquerro, J.A.; Hernandez-Veron, M.A. Domains of global convergence for Newton’s algorithm from auxiliary points. *Appl. Math. Lett.* **2018**, *85*, 48–56. [\[CrossRef\]](#)

30. Ostrowski, A.M. *Solution of Equations in Euclidean and Banach Spaces*; Academic Press: New York, NY, USA, 1973.
31. Cances, E.; Kremlin, G.; Levitt, A. Convergence analysis of direct minimization and self-consistent iterations. *SIAM J. Sci. Comp.* **2021**, *42*, 243–274.
32. De Dterck, H.; He, Y. Anderson acceleration as a Krylov method with application to asymptotic convergence analysis. *arXiv* **2021**, arXiv:2109.14181v1.
33. Rheinboldt, W.C. A unified convergence theory for a class of iterative process. *SIAM J. Numer. Anal.* **1968**, *5*, 42–63. [[CrossRef](#)]
34. Singh, S. A third order iterative algorithm for inversion of cumulative beta distribution. *Numer. Algor.* **2023**, 1–23.
35. Traub, J.F.; Wozdiakowski, H. Convegence and complexity of Newton iteration for operator equations. *J. Assoc. Comput. March.* **1979**, *26*, 250–258. [[CrossRef](#)]

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.