fractal and fractional

Article

Fast and Accurate Numerical Algorithm with Performance Assessment for Nonlinear Functional Volterra Equations

Chinedu Nwaigwe [1,*] and Sanda Micula [2]

1 Department of Mathematics, Rivers State University, Port Harcourt 5080, Nigeria
2 Department of Mathematics, Faculty of Mathematics and Computer Science, Babeş-Bolyai University, Cluj-Napoca, 1 M. Kogălniceanu Street, 400084 Cluj-Napoca, Romania
* Correspondence: nwaigwe.chinedu@ust.edu.ng; Tel.: +23-480-3448-5274

Abstract: An efficient numerical algorithm is developed for solving nonlinear functional Volterra integral equations. The core idea is to define an appropriate operator, then combine the Krasnoselskij iterative scheme with collocation at discrete points and the Newton–Cotes quadrature rule. This results in an explicit scheme that does not require solving a nonlinear or linear algebraic system. For the convergence analysis, the discretization error is estimated and proved to converge via a recurrence relation. The discretization error is combined with the Krasnoselskij iteration error to estimate the total approximation error, hence establishing the convergence of the method. Then, numerical experiments are provided, first, to demonstrate the second order convergence of the proposed method, and secondly, to show the better performance of the scheme over the existing nonlinear-based approach.

Keywords: Krasnoselskij iteration; trapezoidal rule; generalized Banach contraction principle; collocation method; convergence analysis

Citation: Nwaigwe, C.; Micula, S. Fast and Accurate Numerical Algorithm with Performance Assessment for Nonlinear Functional Volterra Equations. *Fractal Fract.* **2023**, 7, 333. https://doi.org/10.3390/fractalfract7040333

Academic Editors: Arsen V. Pskhu, Roman Parovik and Haci Mehmet Baskonus

Received: 12 February 2023
Revised: 12 April 2023
Accepted: 12 April 2023
Published: 17 April 2023

1. Introduction

This study is devoted to analyzing and computing the solution of nonlinear functional Volterra integral equations. These equations have applications in several areas such as physical sciences [1–3], optimal control and economics [4–7], reformulation of more difficult mathematical problems [8,9], and epidemiology [10,11]. In [12], sufficient conditions for the existence of a principal solution were derived for nonlinear Volterra equations and an explicit method was also proposed.

Since closed-form analytical solutions, in general, do not exist, numerical techniques provide a means of approximating them. For example, numerical algorithms have been proposed based on triangular functions [13], collocation methods [14,15], CAS wavelets [16,17], the variational iteration methods [18–21], collocation–trapezoidal methods [22,23], linear programming [24], Picard–trapezoidal rule [9], and Taylor series [25]. Moreover, see [10,11,26–29] for other ideas. Most of these methods are based on directly discretizing the original nonlinear integral equations without using any fixed-point iteration (such as the Picard iteration). In this work, we shall refer to this type of methods as *direct discretization* (DD) algorithms. A typical example is the one proposed in [22].

It is well known that, under suitable conditions and with an arbitrary initial function in a suitable Banach space, the fixed point of an appropriate operator can be approximated using an applicable fixed-point iteration technique, such as the Picard, Krasnoselskij, Mann, or Ishikawa schemes [30–33], see also [34]. These iterative methods can produce analytical expressions for the approximating functions, provided all the operations in the operator are analytically realizable. We shall refer to this type of approach as the *Picard-type* (PT) schemes. See [35] for an example.

The challenge with DD schemes is that they lead to nonlinear algebraic systems which require a lot of computational resources, time, and even high programming skills to solve. The PT schemes face the challenge of not being practically useful once the operations involved in the operator cannot be obtained analytically. This is usually the case in nonlinear problems. Micula [9] came up with the idea of combining PT schemes with DD using the Picard iteration and trapezoidal rule; see also [36] for Mann's iteration.

It is known that the Mann iteration converges faster than the Picard's [36]; however, it is also proved in Theorem 9.4 of [30] that for certain operators, given any Mann iteration that converges to the fixed point, there is always a Krasnoselskij iteration which converges faster. Therefore, the present paper develops the combined technique for functional integral equations using the Krasnoselkij iterative algorithm and a one-dimensional quadrature rule defined at collocation points. The advantages of the approach are as follows: First, unlike the DD schemes, it does not lead to a coupled nonlinear algebraic system, and not even linear systems are encountered. Hence, Newton or other nonlinear solvers are completely avoided and even linear iterative algorithms are also not needed. Second, unlike the PT schemes, every integral is explicitly approximated. A systematic analysis of the convergence of the approach is carried out and numerical examples are provided to show the second-order accuracy of the method. Moreover, the existence and uniqueness of the results in Micula [9,36] are obtained on the basis of a contraction assumption. In the current work, we prove the solvability of the problem without any contraction assumption by employing the generalized Banach contraction principle.

To be precise, the problem investigated in this work is the following nonlinear functional Volterra integral equation of the second kind:

$$u(x) = g(x) + f\left(x, \int_{y=a}^{x} k(x,y,u(y))\,dy\right), \qquad x \in [a,b] \subset, \tag{1}$$

where $C[a,b] \ni g : \to, f : \times \to, k : \times \times \to$.

The solvability of the problem is proved in Section 2, whereas the numerical algorithm which begins with the Krasnoselskij iteration is derived in Section 3. The error and convergence of the method are analyzed in Section 4, whereas numerical examples are provided in Section 5 to demonstrate the accuracy. In Section 6 we assess the performance. Some concluding remarks are given in Section 7.

2. Solvability

We make the following assumptions:

1. $g, k, f(x, \int_{y=a}^{x} k(x,y,u(y))\,dy) \in L^\infty$.
2. The functional f is Lipschitz continuous with respect to the second argument, with a Lipschitz constant $\alpha_f \geq 0$:
$$\|f(x,u) - f(x,v)\| \leq \alpha_f \|u - v\|, \quad \text{for all } u, v \in L^\infty. \tag{2}$$
3. The kernel, k is Lipschitz continuous with respect to u with Lipschitz constant $\alpha_k \geq 0$:
$$\|k(x,y,u) - k(x,y,v)\| \leq \alpha_k \|u - v\| \quad \text{for all } u, v \in L^\infty. \tag{3}$$

We define an operator, T, by

$$(Tu)(x) := g(x) + f\left(x, \int_{y=a}^{x} k(x,y,u(y))\,dy\right). \tag{4}$$

Lemma 1. *The operator T defined in (4) satisfies the inequality:*

$$|T^k u(x) - T^k v(x)| \leq \frac{\left(\alpha_f \alpha_k (x-a)\right)^k}{k!} |u(x) - v(x)|, \ \textit{for } k = 1, 2, \cdots. \tag{5}$$

Proof. We prove this by induction. Setting $k = 1$, we obtain:

$$\begin{aligned}\Big|(Tu)(x) - (Tv)(x)\Big| &= \Big|f\Big(x, \int_a^x k(x,y,u(y))\,dy\Big) - f\Big(x, \int_a^x k(x,y,v(y))\,dy\Big)\Big| \\ &\le \alpha_f \alpha_k \int_a^x \Big|u(y) - v(y)\Big| dy \quad \text{(by Lipschitz Continuity of } f \text{ and } k) \\ &= \alpha_f \alpha_k \Big|u(y) - v(y)\Big| \int_a^x dy \\ &\le \alpha_f \alpha_k (x-a) \Big|u(y) - v(y)\Big|. \end{aligned} \tag{6}$$

This shows that (5) is true for $k = 1$. Next, we let $k = 2$, then we obtain:

$$\begin{aligned}\Big|T^2u(x) - T^2v(x)\Big| &\le \alpha_f \alpha_k \int_a^x \Big|Tu(y) - Tv(y)\Big| dy \\ &\le \alpha_f \alpha_k \int_a^x \alpha_f \alpha_k (y-a)|u(y) - v(y)|\,dy \\ &\le \alpha_f^2 \alpha_k^2 \frac{(x-a)^2}{2} |u(x) - v(x)| \end{aligned}$$

which is also true. Now we assume it is true for any k. Then

$$\begin{aligned}|T^{k+1}u(x) - T^{k+1}v(x)| &\le \alpha_f \alpha_k \int_a^x \Big|T^k u(y) - T^k v(y)\Big| dy \\ &\le \alpha_f \alpha_k \int_a^x \frac{\Big(\alpha_f \alpha_k (y-a)\Big)^k}{k!} |u(y) - v(y)|\,dy \\ &= \frac{\Big(\alpha_f \alpha_k (x-a)\Big)^{k+1}}{(k+1)!} |u(x) - v(x)|. \end{aligned}$$

Hence, it is true for all k. □

Theorem 1 (Solvability)**.** *If Assumptions 1–3 are true, then the nonlinear functional Volterra integral Equation* (1) *has a unique solution.*

Proof. Assumption (1) above guarantees that $T : L^\infty \to L^\infty$. Now, Lemma 1 gives

$$\begin{aligned}|T^k u(x) - T^k v(x)| &\le \frac{\Big(\alpha_f \alpha_k (x-a)\Big)^k}{k!} |u(x) - v(x)|, \text{ for } k = 1, 2, \cdots \\ &\le \frac{\Big(\alpha_f \alpha_k (b-a)\Big)^k}{k!} |u(x) - v(x)| \to 0 \text{ as } k \to \infty. \end{aligned} \tag{7}$$

This shows that T is a contraction and it follows from the generalized Banach contraction principle that T has a unique fixed point which is the unique solution of problem (1). □

3. Numerical Algorithm

This section details the numerical approximation of problem (1). To this end, let $N \in \mathbb{Z}^+$ and define the mesh $\Omega_h = \{x_i = a + ih : i = 0, 1, \cdots, N, h := \frac{b-a}{N}\}$. We also define the grid functions

$$u_i \approx u(x_i) \text{ for each } i, \text{ and } \xi_i^N := h\begin{cases} \frac{1}{2}, \text{ if } i = 0, N, \\ 1, \text{ otherwise,} \end{cases}. \tag{8}$$

Since T is a contraction, the sequence $\{u(x)\}_{n=0}^{\infty}$ generated by

$$u_{n+1}(x) = (1-\lambda)u_n(x) + \lambda T u_n(x), \qquad \lambda \in (0,1), \tag{9}$$

converges to the fixed point of T [30] with the error estimate [31]:

$$\|u_{n+1}(x) - u(x)\| \leq \alpha^n \|u_1(x) - u(x)\| \text{ for each } x \in [a,b], \tag{10}$$

where

$$\alpha = [1 - (1-\hat{\alpha})\lambda] < 1, \, \hat{\alpha} = \alpha_f \alpha_k (b-a) < 1. \tag{11}$$

Lemma 2 (See [37]). *Let $x_i = a + ih, i = 0, N, h = (b-a)/N$ be points in the interval $[a,b]$. Suppose that $f \in C^2[a,b]$. Then*

$$\int_a^b f(x)\,dx = \left(\sum_{j=0}^{N} \left(\xi_j^N f(x_j) \right) \right) + R_f, \tag{12}$$

and there exists $0 \leq R_m < \infty$ such that

$$|R_f| \leq R_m = \max_{\chi \in [a,b]} \left\{ \frac{h^2(b-a)}{12} f''(\chi) \right\} = O(h^2). \tag{13}$$

To derive the method, we first collocate problem (1) at $x_0 = a \in \Omega_h$. This gives:

$$u(x_0) = g(x_0) + f(x_0, 0). \tag{14}$$

Observe that this is exact as no approximation has been made. Hence, we can initialize the Krasnoselskij sequence (9) from (14) as follows:

$$u_0(x) = g(x) + f(x,0), \text{ for all } x \in [a,b].$$

Therefore, the iteration becomes:

$$\begin{cases} u_{n+1}(x) = (1-\lambda)u_n(x) + \lambda (T u_n)(x), & n \geq 0, \\ u_0(x) = g(x) + f(x,0), & 0 < \lambda < 1. \end{cases} \tag{15}$$

We now collocate (15) at $x_i \in \Omega_h \setminus \{x_0\}$:

$$\begin{aligned} &0 < \lambda < 1, \qquad u_0(x_i) = g(x_i) + f(x_i, 0), \\ &n \geq 0 : \begin{cases} I_n(x_i) = \int_{y=a}^{x_i} k(x_i, y, u_n(y))\,dy, \\ u_{n+1}(x_i) = (1-\lambda)u_n(x_i) + \lambda(g(x_i) + f(x_i, I_n(x_i))). \end{cases} \end{aligned} \tag{16}$$

Using the trapezoidal rule in Lemma 2 to approximate the integral on the iterative scheme (16), we have the algorithm:

$$\begin{cases} & x_i \in \Omega_h \setminus \{x_0\}, n = 0, 1, \cdots, \\ \hat{u}_{n+1}(x_i) & = (1-\lambda)\hat{u}_n(x_i) + \lambda\left(g(x_i) + f(x_i, \hat{I}_{h,n}(x_i))\right), \\ \hat{u}_0(x_i) & = u_0(x_i), \quad \text{for all } x_i \in \Omega_h, \\ \hat{I}_{h,n}(x_i) & = \sum_{j=0}^{i} \xi_j^i k(x_i, x_j, \hat{u}_n(x_j)). \end{cases} \tag{17}$$

The system (17) constitutes the numerical algorithm for approximating problem (1). We prove in the next section that a sequence of solutions computed with this scheme converges to the exact solution of problem (1).

4. Convergence Analysis

Definition 1 (Maximum Error)**.** *The numerical error, e_h, is the maximum error committed in approximating $u(x_i)$ by using the scheme (17) for all $x_i \in \Omega_h$ when $n \to \infty$. That is*

$$e_h = \lim_{n\to\infty}\left(\max_{x_i\in\Omega_h} e_n(x_i)\right) \tag{18}$$

where

$$e_n(x_i) = |u(x_i) - \hat{u}_n(x_i)|.$$

Remark 1. *The goal in this section is to show that the quantity $e_n(x_i)$ vanishes whenever $n \to \infty$, $h \to 0$, and assumption (11) holds.*

Let us first prove the following lemma.

Lemma 3. *Let γ be a Lipschitz operator with constant α_g. Then*

$$\gamma(u+v) \le \gamma(u) + \alpha_g|v| \textit{ for all } u, v \in Dom(\gamma). \tag{19}$$

Proof. The result is trivial when $v = 0$. It is also trivial if $v \neq 0$ and $\gamma(u+v) \le \gamma(u)$. We only prove the inequality for the case when $v \neq 0$ and

$$\gamma(u+v) > \gamma(u). \tag{20}$$

By the Lipschitz continuity of γ, we have

$$|\gamma(u+v) - \gamma(u)| \le \alpha_g|v|.$$

Because of (20), we can write the left side of the last inequality as:

$$\gamma(u+v) - \gamma(u) \le \alpha_g|v|.$$

Hence the result. □

Lemma 4 (Recurrence Relation)**.** *The error, $\hat{R}_n(x_i) = |u_n(x_i) - \hat{u}_n(x_i)|$, committed in using the scheme (17) to approximate the iterative process (16) satisfies the recurrence relation:*

$$\hat{R}_{n+1}(x_i) \le \left(1 - \lambda + \lambda\alpha_f\alpha_k\sum_{j=1}^{i}\xi_j^i\right)\hat{R}_n(x_i) + \lambda\alpha_f|R_m|. \tag{21}$$

Proof. First, since $\hat{u}_0(x_i) = u_0(x_i)$, it means that $\hat{R}_0 = 0$.
Setting $n = 0$ in (17), we have:

$$\begin{aligned}
u_1(x_i) &= (1-\lambda)u_0(x_i) + \lambda\left(g(x_i) + f\left(x_i, \sum_{j=0}^{i}\xi_j^i k(x_i, x_j, u_0(x_j)) + \tilde{R}_1\right)\right) \\
&\le (1-\lambda)u_0(x_i) + \lambda\left(g(x_i) + f\left(x_i, \sum_{j=0}^{i}\xi_j^i k(x_i, x_j, u_0(x_j))\right) + \alpha_f|\tilde{R}_1|\right) \quad \text{(see Lemma 3)} \\
&\le (1-\lambda)u_0(x_i) + \lambda\left(g(x_i) + f\left(x_i, \sum_{j=0}^{i}\xi_j^i k(x_i, x_j, u_0(x_j))\right)\right) + \alpha_f\lambda|R_m| \\
&= \hat{u}_1(x_i) + \alpha_f\lambda|R_m|.
\end{aligned} \tag{22}$$

Hence,

$$\begin{aligned} \hat{R}_1 &:= |u_1(x_i) - \hat{u}_1(x_i)| \\ &\leq \alpha_f \lambda |R_m| = \left(1 - \lambda + \lambda \alpha_f \alpha_k \sum_{j=1}^{n-1} \xi_j^i \right) \hat{R}_0(x_i) + \alpha_f \lambda |R_m|. \end{aligned} \tag{23}$$

Similarly, setting $n = 1$, we obtain:

$$\begin{aligned} &u_2(x_i) = (1-\lambda)u_1(x_i) + \lambda\left(g(x_i) + f\left(x_i, \sum_{j=0}^{i} \xi_j^i k(x_i, x_j, u_1(x_j)) + \tilde{R}_2\right)\right) \\ &\leq (1-\lambda)\hat{u}_1(x_i) + (1-\lambda)\hat{R}_1(x_i) \\ &+ \lambda\left(g(x_i) + f\left(x_i, \sum_{j=0}^{i} \xi_j^i k\left(x_i, x_j, \hat{u}_1(x_j) + \hat{R}_1\right) + R_m\right)\right) \\ &\leq (1-\lambda)\hat{u}_1(x_i) + \lambda\left(g(x_i) + f\left(x_i, \sum_{j=0}^{i} \xi_j^i k(x_i, x_j, \hat{u}_1(x_j)) + \alpha_k |\hat{R}_1| \sum_{j=0}^{i} \xi_j^i + R_m\right)\right) \\ &+ (1-\lambda)\hat{R}_1(x_i) \\ &\leq (1-\lambda)\hat{u}_1(x_i) + \lambda\left(g(x_i) + f\left(x_i, \sum_{j=0}^{i} \xi_j^i k(x_i, x_j, \hat{u}_1(x_j))\right)\right) \\ &+ \left(1 - \lambda + \lambda\alpha_f\alpha_k \sum_{j=1}^{i} \xi_j^i\right)\hat{R}_1(x_i) + \lambda\alpha_f|R_m| \\ &= \hat{u}_2(x_i) + \left(1 - \lambda + \lambda\alpha_f\alpha_k \sum_{j=1}^{i} \xi_j^i\right)\hat{R}_1(x_i) + \lambda\alpha_f|R_m|. \end{aligned} \tag{24}$$

Hence,

$$\begin{aligned} \hat{R}_2(x_i) &= |u_2(x_i) - \hat{u}_2(x_i)| \\ &\leq \left(1 - \lambda + \lambda\alpha_f\alpha_k \sum_{j=1}^{i} \xi_j^i\right)\hat{R}_1(x_i) + \lambda\alpha_f|R_m|. \end{aligned} \tag{25}$$

In general,

$$\begin{aligned} &u_{n+1}(x_i) = (1-\lambda)u_n(x_i) + \lambda\left(g(x_i) + f\left(x_i, \sum_{j=0}^{i} \xi_j^i k(x_i, x_j, u_n(x_j)) + \tilde{R}_{n+1}\right)\right) \\ &\leq (1-\lambda)\hat{u}_n(x_i) + (1-\lambda)\hat{R}_n(x_i) \\ &+ \lambda\left(g(x_i) + f\left(x_i, \sum_{j=0}^{i} \xi_j^i k(x_i, x_j, \hat{u}_n(x_j) + \hat{R}_n) + R_m\right)\right) \\ &\leq (1-\lambda)\hat{u}_n(x_i) + \lambda\left(g(x_i) + f\left(x_i, \sum_{j=0}^{i} \xi_j^i k(x_i, x_j, \hat{u}_n(x_j)) + \alpha_k \hat{R}_n \sum_{j=0}^{i} \xi_j^i + R_m\right)\right) \\ &+ (1-\lambda)\hat{R}_n(x_i) \\ &\leq (1-\lambda)\hat{u}_n(x_i) + \lambda\left(g(x_i) + f\left(x_i, \sum_{j=0}^{i} \xi_j^i k(x_i, x_j, \hat{u}_n(x_j))\right)\right) \\ &+ \left(1 - \lambda + \lambda\alpha_f\alpha_k \sum_{j=1}^{i} \xi_j^i\right)\hat{R}_n(x_i) + \lambda\alpha_f|R_m| \\ &= \hat{u}_{n+1}(x_i) + \left(1 - \lambda + \lambda\alpha_f\alpha_k \sum_{j=1}^{i} \xi_j^i\right)\hat{R}_n(x_i) + \lambda\alpha_f|R_m|. \end{aligned} \tag{26}$$

Inequality (26) gives:

$$\hat{R}_{n+1} := |u_{n+1}(x_i) - \hat{u}_{n+1}(x_i)| \tag{27}$$

$$\leq \left(1 - \lambda + \lambda\alpha_f\alpha_k \sum_{j=1}^{i} \xi_j^i\right)\hat{R}_n(x_i) + \lambda\alpha_f|R_m|. \tag{28}$$

This proves the claim. □

Theorem 2 (Convergence). *The error $\hat{R}_n(x_i) = |u_n(x_i) - \hat{u}_n(x_i)|$ committed in using the scheme (17) to approximate the iterative process (16) satisfies:*

$$\hat{R}_n(x_i) \leq c_2 \sum_{j=0}^{n-1} c_1^j, \tag{29}$$

hence

$$\lim_{n\to\infty} \hat{R}_n(x_i) \leq \frac{\alpha_f}{1 - \alpha_f\alpha_k \sum_{j=0}^{i} \xi_j^i}|R_m| = O(h^2), \tag{30}$$

where $c_1 = 1 - \lambda + \lambda\alpha_f\alpha_k \sum_{j=1}^{n-1} \xi_j^i$ and $c_2 = \lambda\alpha_f|R_m|$.

Proof. From Lemma 21, we have

$$\begin{aligned}
\hat{R}_n(x_i) &\leq \left(1 - \lambda + \lambda\alpha_f\alpha_f \sum_{j=1}^{i} \xi_j^i\right)\hat{R}_{n-1}(x_i) + \lambda\alpha_f|R_m| \\
&= c_1\hat{R}_{n-1} + c_2 \leq c_1\left(c_1\hat{R}_{n-2} + c_2\right) + c_2 \\
&\leq c_1^2\hat{R}_{n-2} + c_1c_2 + c_2 \\
&\vdots \\
&\leq c_1^n\hat{R}_0 + c_2\sum_{j=0}^{n-1} c_1^j \\
&= c_2\sum_{j=0}^{n-1} c_1^j.
\end{aligned} \tag{31}$$

Taking limit:

$$\lim_{n\to\infty} \hat{R}_n \leq \frac{c_2}{1 - c_1} = \frac{\alpha_f}{1 - \alpha_f\alpha_k \sum_{j=0}^{i} \xi_j^i}|R_m| = O(h^2). \tag{32}$$

□

Theorem 3 (Convergence). *The numerical solution, $\hat{u}_n(x)$, computed using the scheme (17), converges to the exact solution, $u(x)$.*

Proof. The error $e_n(x_i)$ between $u(x_i)$ and $\hat{u}_n(x_i)$ is

$$\begin{aligned}
e_n(x_i) &= |u(x_i) - \hat{u}_n(x_i)| = |u(x_i) - u_n(x_i) + u_n(x_i) - \hat{u}_n(x_i)| \\
&\leq |u(x_i) - u_n(x_i)| + |u_n(x_i) - \hat{u}_n(x_i)| \\
&\leq \alpha^n|u(x_i) - u_n(x_i)| + \frac{\alpha_f}{1 - \alpha_f\alpha_k \sum_{j=0}^{i} \xi_j^i}|R_m| \\
&= \alpha^n|u(x_i) - u_n(x_i)| + O(h^2).
\end{aligned}$$

where $\alpha < 1$ is defined in (11). Hence, the convergence result follows (see definition 1 and the remark that follows it). □

Remark 2. *Observe the appearance of* $1 - \alpha_f \alpha_k \sum_{j=0}^{i} \xi_j^i$ *in* (30). *This implies that we require this quantity to be non-negative (since the left hand side of* (30) *is positive). Since* $\alpha_f \alpha_k \sum_{j=0}^{i} \xi_j^i \leq \alpha_f \alpha_k (b-a)$, *it follows that the requirement is satisfied if*

$$\alpha_f \alpha_k (b-a) < 1. \tag{33}$$

This is a requirement for the scheme to be convergent.

Remark 3 (Implication of the analysis)**.** *As observed in the proof of Theorem 3, the numerical error consists of two parts—the fixed-point iteration error and the quadrature error. Hence, both errors must converge to zero for the proposed method to converge to the exact solution, and the solutions computed with the method would be obtained at minimal computational cost compared to methods which involve solving nonlinear systems. However, since the convergence of the fixed-point iteration is guaranteed whenever the operator is a contraction, it then implies that if an appropriate quadrature rule is in place, then the contractivity of the associated operator guarantees that numerical solutions can be computed accurately and efficiently.*

5. Numerical Experiments

Numerical examples are now presented to assess the accuracy and efficiency of the scheme (17). The examples are derived through the method of manufacture solutions [38]. In each of the four problems, a sequence of solutions is computed with different meshes of varying sizes. In all the computations, we take $\lambda = 0.5$ and terminate each Krasnoselskij iteration whenever $|u_{n+1}(x_i) - u_n(x_i)| \leq 8 \times 10^{-11}$. The error (in maximum norm) is computed as in (18), whereas the experimental order of convergence is computed using:

$$EOC = \frac{log\left(\frac{e_h}{e_{h/2}}\right)}{log(2)}, \tag{34}$$

see [39].

5.1. Example 1

The following problem is considered (see [22]):

$$u(x) = g(x) + \frac{\sin(x)}{1 + \left(\int_0^x u(y) e^{x-y-u^2} dy\right)^2}, \quad x \in [0,1]. \tag{35}$$

where

$$g(x) = e^{-x} - \frac{\sin(x)}{\left(-e^{-1}e^x/2 + e^x e^{-e^{-2x}}/2\right)^2 + 1}.$$

The exact solution of this problem is $u(x) = e^{-x}$. Table 1 tabulates the numerical results and shows that the computed solution converges to the exact solution with second order of accuracy. Figure 1 shows the plots of the numerical and exact solutions on grids with 2, 3, 6 and 20 grid points. The convergence of the method is obviously ascertained.

Table 1. Numerical Results for Example 1. N = number of sub-intervals. Errors are computed with the infinity norm. EOC = Experimental order of convergence.

N	Error	EOC
1	0.01213743114921989	-
2	0.005090501313527285	1.2535834665481371
4	0.0013464449826251501	1.9186524594014347
8	0.00035011541689816683	1.943252785210336
16	8.790389104307295 $\times 10^{-5}$	1.993831658033252
32	2.2059673406571445 $\times 10^{-5}$	1.9945155953164302
64	5.515784523291156 $\times 10^{-6}$	1.9997734284826107
128	1.3789984285583756 $\times 10^{-6}$	1.9999452855688975
256	3.4476374671799093 $\times 10^{-7}$	1.9999408304436097
512	8.619000313458969 $\times 10^{-8}$	2.000015626096606
1024	2.1545896289332234 $\times 10^{-8}$	2.000107431585809
2048	5.385888846021203 $\times 10^{-9}$	2.000156753556326

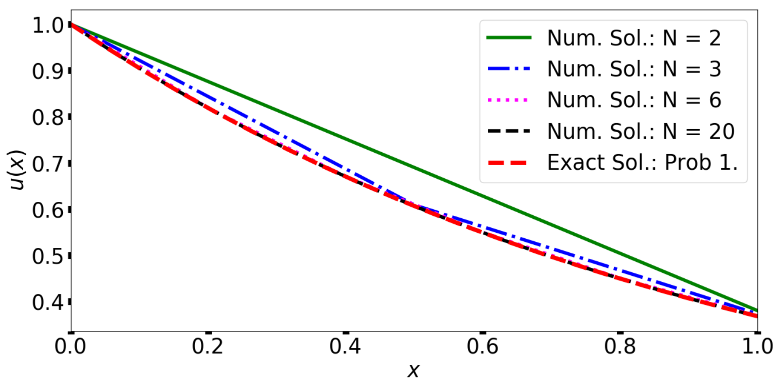

Figure 1. Plot of solution of Example 1 for different grid sizes.

5.2. Example 2

In this example, we consider the problem (see also [22]):

$$u(x) = -\frac{x^2}{9} - \left(x^{6/27} + 1\right)^{1/3} + f\left(x, \int_0^1 k(x, y, u(y))dy\right), \tag{36}$$

where

$$f(x, I) = (I^2 + 1)^{1/3}, \; k(x, y, u) = \begin{cases} u, & \text{if } y \leq x, \\ 0, & \text{otherwise.} \end{cases}$$

The exact solution to this problem is $u(x) = \frac{x^2}{9}$. The numerical results as tabulated in Table 2 show that the computed solution converges to the exact solution with second order of accuracy. To make the discussion easier to understand, the solutions computed on different grids are plotted in Figure 2, and the convergence of the method is obvious.

Table 2. Numerical results for Example 2.

N	Error	EOC
1	0.0005814856077918928	-
2	0.00012230339333171858	2.2492790520697135
4	2.91619860433856 $\times 10^{-5}$	2.0683035510752483
8	7.202538943651415 $\times 10^{-6}$	2.017511515015079
16	1.7951203420268902 $\times 10^{-6}$	2.0044249926801028
32	4.4841246069071694 $\times 10^{-7}$	2.001182289199546
64	1.1207335620655456 $\times 10^{-7}$	2.000383029609112
128	2.80046026646108 $\times 10^{-8}$	2.000707475089127
256	6.996875440146155 $\times 10^{-9}$	2.0008812453274856
512	1.7429708232263863 $\times 10^{-9}$	2.005162389306972
1024	4.336317416253621 $\times 10^{-10}$	2.0070061491915854
2048	1.0526418625644851 $\times 10^{-10}$	2.042455689416187
4096	2.525327169600189 $\times 10^{-11}$	2.059472461771361

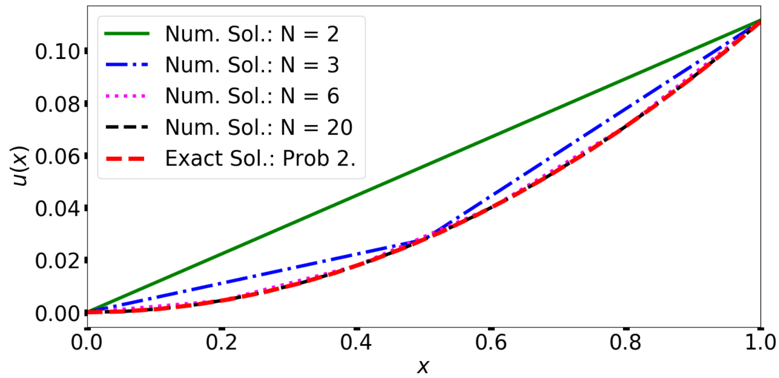

Figure 2. Plot of solution of Example 2 for different grid sizes.

5.3. Example 3

As a third example, we consider the problem:

$$u(x) = g(x) + \frac{x^2}{1 + \left(\int_0^x \frac{x^3 y^5}{1+u^2} dy\right)^2} \quad x \in [0,1]. \tag{37}$$

where

$$g(x) = \frac{x^2(x(x^6 log(x^6+1)^2 + 36) - 36)}{x^6 log(x^6+1)^2 + 36}.$$

The exact solution of this problem is $u(x) = x^3$. It can also be seen in Table 3 that the numerical solution converges to the exact solution with second order of convergence. Moreover, Figure 3 demonstrates the convergence of the method as the grid is refined from just two points to twenty points. The proposed numerical scheme is clearly convergent. The astonishing feature of the method is that very high accuracy is attained even with very coarse grids.

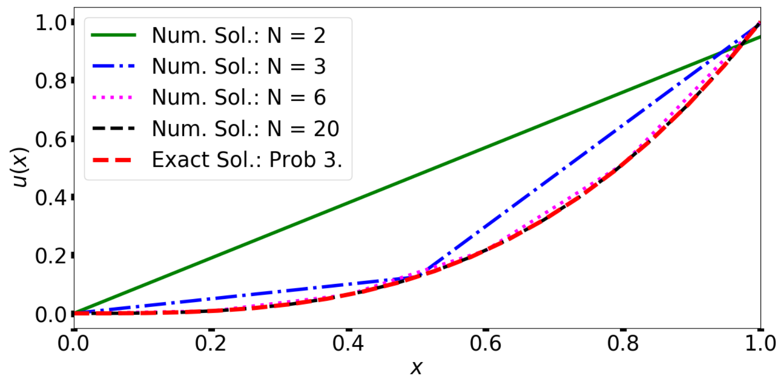

Figure 3. Plot of solution of Example 3 for different grid sizes.

Table 3. Numerical results for Example 3. Results show second-order convergence.

N	Error	EOC
1	0.05163523545962034	-
2	0.006373161885883882	3.018274672426548
4	0.0012307613112914062	2.372458308685062
8	0.0002983915555665462	2.044272385454718
16	$7.400954062197762 \times 10^{-5}$	2.0114235414553514
32	$1.8465882868468064 \times 10^{-5}$	2.002849020979508
64	$4.6142027866347135 \times 10^{-6}$	2.000708925979108
128	$1.1534254369394148 \times 10^{-6}$	2.00015666531903
256	$2.883532977948633 \times 10^{-7}$	2.0000153169770556
512	$7.209643260175369 \times 10^{-8}$	1.9998377416249797
1024	$1.802702365161224 \times 10^{-8}$	1.9997666548103157
2048	$4.511132578599586 \times 10^{-9}$	1.9985996290821917
4096	$1.129273896616212 \times 10^{-9}$	1.998094243306733
8192	$2.8453139844231146 \times 10^{-10}$	1.9887356731658479

5.4. Example 4

Finally, we consider the problem:

$$u(x) = g(x) + x^3 + \frac{1}{3}\left[\frac{1}{4}\left(\int_0^x k(x,y,u(y))dy\right)^4 - \frac{3}{2}\left(\int_0^x k(x,y,u(y))dy\right)^2 + \int_0^x k(x,y,u(y))dy\right] \quad x \in [0,1],$$

where

$$k(x,y,u(y)) = x^4 + y^2 - \frac{1}{3}u^3(y) + u^2(y)$$

and

$$g(x) = -0.333333333333333x^5 - 1.11111111111111x^3$$
$$- 0.166666666666667x - 0.0833333333333333(x^5 + x^3/3 + x/2+$$
$$\sin(x)^3/9 + \sin(x)\cos(x)/2 - \sin(x)/3)^4 + 0.5(x^5 + x^3/3 + x/2$$
$$+ \sin(x)^3/9 + \sin(x)\cos(x)/2 - \sin(x)/3)^2$$
$$- 0.037037037037037\sin(x)^3 - 0.166666666666667\sin(x)\cos(x)$$
$$+ 0.111111111111111\sin(x) + \cos(x).$$

The exact solution to this problem is $u(x) = \cos(x)$. Similar to the previous examples, one can see in Table 4 that the numerical solution converges to the exact solution with second order of convergence. Figure 4 displays the plots of the numerical and exact solutions on different grids. It is evident that the method converges.

Table 4. Numerical results for Example 4. Results show second-order convergence.

N	Error	EOC
1	0.12721182356988314	-
2	0.019477425482774535	2.707357866274032
4	0.004206018337715833	2.211275950082332
8	0.0010008191561557966	2.0712738313024626
16	0.0002468037760221531	2.019744936190632
32	$6.148467592825835 \times 10^{-5}$	2.0050656757391567
64	$1.5357610263277977 \times 10^{-5}$	2.0012731451169947
128	$3.838580611259523 \times 10^{-6}$	2.000308890953441
256	$9.596195429395493 \times 10^{-7}$	2.0000385014534188
512	$2.3991099107334435 \times 10^{-7}$	1.999963285326972
1024	$5.999148333657445 \times 10^{-8}$	1.9996696446899704
2048	$1.5002635800343 \times 10^{-8}$	1.999541714861857

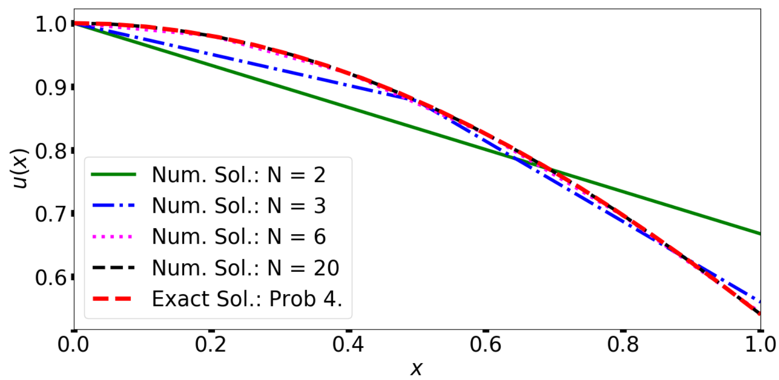

Figure 4. Plot of solution of Example 4 for different grid sizes.

6. Performance Analysis

In this section, we reuse the examples in Section 5 to assess the computational efficiency of the proposed method in comparison with a nonlinear system based (DD) method as proposed in [22]. We achieve this by comparing CPU times used by each method. We will also briefly discuss the memory usage of the scheme. For example 1, we set $\lambda = 0.9$; for example 2, we set $\lambda = 0.8$; example 3 used $\lambda = 0.9$; whereas example 4 used $\lambda = 0.95$. Both algorithms (the proposed method and that of [22]) solve each of the four example problems on $N = 4096$ equal sub-intervals in $[0, 1]$. Tables 5–8 display the results with the elapsed CPU time and error committed by each method in solving the problem. It is obvious that the proposed method highly outperforms the direct discretization (DD) method [22] as it has much better computational efficiency. It is important to notice that the error of the DD scheme is not better than that of the new scheme.

In addition to the above merits, the new scheme is also more memory efficient than the DD scheme since it does not require to solve linear or nonlinear systems. Yet, from the programming point of view, the new scheme is very easy to program or implement. All these advantages lead to the conclusion that the present method is highly competitive.

Table 5. Performance results from Example 1. (Proposed scheme used $\lambda = 0.9$).

Scheme	CPU Time (s)	Error
Proposed Method	210.16	1.347×10^{-9}
Direct Discretization	839.41	1.347×10^{-9}

Table 6. Performance results from Example 2. (Proposed scheme used $\lambda = 0.8$).

Scheme	CPU Time (s)	Error
Proposed Method	72.24	2.714×10^{-11}
Direct Discretization	144.52	2.738×10^{-11}

Table 7. Performance results from Example 3. (Proposed scheme used $\lambda = 0.9$).

Scheme	CPU Time (s)	Error
Proposed Method	122.89	1.127×10^{-9}
Direct Discretization	264.84	1.127×10^{-9}

Table 8. Performance results from Example 4. (Proposed scheme used $\lambda = 0.95$).

Scheme	CPU Time (s)	Error
Proposed Method	149.10	3.75×10^{-9}
Direct Discretization	189.18	3.75×10^{-9}

7. Conclusions

A fixed-point iteration—the Krasnoselskij scheme—is used to construct an efficient scheme for solving nonlinear functional Volterra equations without a need for nonlinear systems nor a concern for differentiability of the kernels or functionals. The convergence to the exact solution is thoroughly analyzed. Numerical experiments are provided and lead to the following conclusions:

1. The scheme is very accurate and competitive;
2. The scheme is less computationally expensive than the direct discretization methods, such as the one proposed in [22];
3. It is also more memory efficient than that of [22];
4. This approach should be adopted whenever possible.

The implication is that if the associated operator is a contraction (satisfying a condition similar to inequality (33)), then fixed-point methods can be formulated to approximate the solution accurately and efficiently. As a further study, we intend to investigate the convergence and performance of other iterative algorithms for related operator equations.

Author Contributions: Conceptualization, C.N. and S.M.; methodology, C.N. and S.M.; software, C.N.; validation, C.N. and S.M.; formal analysis, C.N.; investigation, C.N.; resources, C.N.; writing—original draft preparation, C.N. and S.M.; writing—review and editing, C.N. and S.M.; visualization, C.N.; supervision, C.N. and S.M.; project administration, C.N.; funding acquisition, S.M. 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.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the reviewers and editors for their valuable comments, which improved the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Abdou, M. On a symptotic methods for Fredholm–Volterra integral equation of the second kind in contact problems. *J. Comput. Appl. Math.* **2003**, *154*, 431–446. [CrossRef]
2. Le, T.D.; Moyne, C.; Murad, M.; Lima, S. A two-scale non-local model of swelling porous media incorporating ion size correlation effects. *J. Mech. Phys. Solids* **2013**, *61*, 2493–2521. [CrossRef]
3. Rocha, A.C.; Murad, M.A.; Moyne, C.; Oliveira, S.P.; Le, T.D. A new methodology for computing ionic profiles and disjoining pressure in swelling porous media. *Comput. Geosci.* **2016**, *20*, 975–996. [CrossRef]
4. Hu, S.; Khavanin, M.; Zhuang, W. Integral equations arising in the kinetic theory of gases. *Appl. Anal.* **1989**, *34*, 261–266. [CrossRef]
5. Jerri, A. *Introduction to Integral Equations with Applications*; John Wiley & Sons: Hoboken, NJ, USA, 1999.
6. Oregan, D. Existence results for nonlinear integral equations. *J. Math. Anal. Appl.* **1995**, *192*, 705–726. [CrossRef]
7. Prüss, J. *Evolutionary Integral Equations and Applications*; Birkhäuser: Basel, Switzerland, 2013; Volume 87.
8. Wazwaz, A.M. *First Course In Integral Equations, A*; World Scientific Publishing Company: Singapore, 2015.
9. Micula, S. On Some Iterative Numerical Methods for Mixed Volterra–Fredholm Integral Equations. *Symmetry* **2019**, *11*, 1200. [CrossRef]
10. Maleknejad, K.; Hadizadeh, M. A new computational method for Volterra-Fredholm integral equations. *Comput. Math. Appl.* **1999**, *37*, 1–8. [CrossRef]
11. Wazwaz, A.M. A reliable treatment for mixed Volterra–Fredholm integral equations. *Appl. Math. Comput.* **2002**, *127*, 405–414. [CrossRef]
12. Sidorov, D. Existence and blow-up of Kantorovich principal continuous solutions of nonlinear integral equations. *Differ. Equ.* **2014**, *50*, 1217–1224. [CrossRef]
13. Manochehr, K. Triangular functions for numerical solution of the nonlinear Volterra Integral Equations. *J. Appl. Math. Comput.* **2021**, *68*, 1979–2002.
14. Ordokhani, Y.; Razzaghi, M. Solution of nonlinear Volterra–Fredholm–Hammerstein integral equations via a collocation method and rationalized Haar functions. *Appl. Math. Lett.* **2008**, *21*, 4–9. [CrossRef]
15. Brunner, H. On the numerical solution of nonlinear Volterra–Fredholm integral equations by collocation methods. *SIAM J. Numer. Anal.* **1990**, *27*, 987–1000. [CrossRef]
16. Ezzati, R.; Najafalizadeh, S. Numerical methods for solving linear and nonlinear Volterra-Fredholm integral equations by using CAS wavelets. *World Appl. Sci. J.* **2012**, *18*, 1847–1854.
17. Shiralashetti, S.; Lamani, L. CAS Wavelets Stochastic Operational Matrix of Integration and its Application for Solving Stochastic Itô-Volterra Integral Equations. *Jordan J. Math. Stat.* **2021**, *14*, 555–580.
18. Xu, L. Variational iteration method for solving integral equations. *Comput. Math. Appl.* **2007**, *54*, 1071–1078. [CrossRef]
19. Sheth, S.S.; Singh, D. An Analytical Approximate Solution of Linear, System of Linear and Non Linear Volterra Integral Equations Using Variational Iteration Method. In Proceedings of the International Conference on Advancements in Computing & Management (ICACM), Jaipur, India, 13–14 April 2019.
20. Yousefi, S.A.; Lotfi, A.; Dehghan, M. He's variational iteration method for solving nonlinear mixed Volterra–Fredholm integral equations. *Comput. Math. Appl.* **2009**, *58*, 2172–2176. [CrossRef]
21. Hamoud, A.; Ghadle, K. On the numerical solution of nonlinear Volterra-Fredholm integral equations by variational iteration method. *Int. J. Adv. Sci. Tech. Res.* **2016**, *3*, 45–51.
22. Bazm, S.; Lima, P.; Nemati, S. Analysis of the Euler and trapezoidal discretization methods for the numerical solution of nonlinear functional Volterra integral equations of Urysohn type. *J. Comput. Appl. Math.* **2021**, *398*, 113628. [CrossRef]
23. Nwaigwe, C.; Benedict, D.N. Generalized Banach fixed-point theorem and numerical discretization for nonlinear Volterra–Fredholm equations. *J. Comput. Appl. Math.* **2023**, *425*, 115019. [CrossRef]

24. Hasan, P.M.; Suleiman, N. Numerical Solution of Mixed Volterra-Fredholm Integral Equations Using Linear Programming Problem. *Appl. Math.* **2018**, *8*, 42–45.
25. Chen, Z.; Jiang, W. An approximate solution for a mixed linear Volterra–Fredholm integral equation. *Appl. Math. Lett.* **2012**, *25*, 1131–1134. [CrossRef]
26. Atkinson, K.E. The numerical solution of integral equations of the second kind. In *Cambridge Monographs on Applied and Computational Mathematics*; Cambridge University Press: Cambridge, UK, 1996.
27. Aziz, I.; ul Islam, S. New algorithms for the numerical solution of nonlinear Fredholm and Volterra integral equations using Haar wavelets. *J. Comput. Appl. Math.* **2013**, *239*, 333–345. [CrossRef]
28. Nwaigwe, C. Solvability and Approximation of Nonlinear Functional Mixed Volterra–Fredholm Equation in Banach Space. *J. Integral Equ. Appl.* **2022**, *34*, 489–500. [CrossRef]
29. Youssri, Y.; Hafez, R. Chebyshev collocation treatment of Volterra–Fredholm integral equation with error analysis. *Arab. J. Math.* **2020**, *9*, 471–480. [CrossRef]
30. Berinde, V. *Iterative Approximation of Fixed Points*; Springer: Berlin/Heidelberg, Germany, 2007.
31. Okeke, G.A. Convergence analysis of the Picard–Ishikawa hybrid iterative process with applications. *Afr. Mat.* **2019**, *30*, 817–835. [CrossRef]
32. Ofem, A.; Igbokwe, D.; Udo-utun, X. Implicit iteration process for Lipschitzian α-hemicontraction semigroups. *MathLAB J.* **2020**, *7*, 43–52.
33. Liou, Y.C. Computing the fixed points of strictly pseudocontractive mappings by the implicit and explicit iterations. *Abstr. Appl. Anal.* **2012**, *2012*, 315835. [CrossRef]
34. Osilike, M. Nonlinear accretive and pseudo-contractive operator equations in Banach spaces. *Nonlinear Anal.* **1996**, *31*, 779–789.
35. Hasan, P.M.; Sulaiman, N.A.; Soleymani, F.; Akgül, A. The existence and uniqueness of solution for linear system of mixed Volterra-Fredholm integral equations in Banach space. *AIMS Math.* **2020**, *5*, 226–235. [CrossRef]
36. Micula, S. Numerical Solution of Two-Dimensional Fredholm–Volterra Integral Equations of the Second Kind. *Symmetry* **2021**, *13*, 1326. [CrossRef]
37. Burden, R.L. *Numerical Analysis*; Cengage Learning': Boston, MA, USA, 2011.
38. Nwaigwe, C. An Unconditionally Stable Scheme for Two-Dimensional Convection-Diffusion-Reaction Equations. 2022. Available online: https://www.researchgate.net/publication/357606287_An_Unconditionally_Stable_Scheme_for_Two-Dimensional_Convection-Diffusion-Reaction_Equations (accessed on 11 February 2023).
39. Nwaigwe, C. Coupling Methods for 2d/1d Shallow Water Flow Models for Flood Simulations. Ph.D. Thesis, University of Warwick, Coventry, UK, 2016.