Int. J. Nonlinear Anal. Appl. 12 (2021) No. 2, 237-248
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2020.18371.2007

Newton-Taylor polynomial solutions of systems of nonlinear differential equations with variable coefficients

Bahman Babayar-Razlighi[a,*]

[a]*Department of Mathematics, Qom University of Technology, Qom, Iran*

(Communicated by Reza Saadati)

Abstract

The main purpose of this paper is consider Newton-Taylor polynomial solutions method in numerical solution of nonlinear system of differential equations. We apply Newton's method to linearize it. We found Taylor polynomial solution of the linear form. Sufficient conditions for convergence of the numerical method are given and their applicability is illustrated with some examples. In numerical examples we give two benchmark sample problems and compare the proposed method by the famous Runge-Kutta fourth-order method. These sample problems practically show some advantages of the Newton-Taylor polynomial solutions method.

Keywords: Variable coefficients, Newton's method, Taylor polynomial solutions, Ordinary differential equations, Nonlinear systems.
2020 MSC: 34L30, 65D15, 65L05.

---

1. Introduction

In the present paper, we consider the Newton-Taylor polynomial solutions method for a system of nonlinear differential equations of the form

$$\begin{cases} U'(t) = f(t, U(t)),\ 0 \le t, \\ U(0) = U_0. \end{cases} \tag{1.1}$$

*Corresponding author
Email address: babayar@qut.ac.ir (Bahman Babayar-Razlighi)

Received: July 2019 *Accepted:* February 2020

where $U(t) = (u_1(t), ..., u_d(t))^T$is the unknown vector of functions and $U_0 = \left(u_1^{(0)}, ..., u_d^{(0)}\right)^T$ is the known initial vector, for some $d \in \mathbb{N}$,

$$U^{'}(t) := (u_1^{'}(t), ..., u_d^{'}(t))^T, \tag{1.2}$$

$$f(t, U(t)) := (f_1(t, U(t)), ..., f_d(t, U(t))^T, \tag{1.3}$$

Here f is known function of their arguments and in this paper, $lhs := rhs$, means that lhs is defined by the rhs. Also this equations have been solved by different numerical techniques. Some of these methods are direct methods such as Runge-Kutta method [7], and some of them are applicable when they reduces to a system of Volterra integral equations such as Newton-Product method [2], extrapolation method [3], and another Volterra integral representations methods [13, 6, 11, 9], for some Various f.

According to the chapter two of [8] there exists a maximal $b > 0$ such that the equation (1.1) has a unique solution on $[0, b]$. Rewrite Eq. (1.1) on $[0, b]$ as follow

$$\begin{cases} U^{'}(t) - f(t, U(t)) = 0, \ 0 \le t \le b, \\ U(0) = U_0. \end{cases} \tag{1.4}$$

We shall apply the method on (1.4). In Section 2, we apply the Newton's method to linearize (1.4). A convergence analysis of Newton's method for the problem is provided in the Subsections of Section 2. Same convergence analysis are given in [4] for a Stefan Problem with Volterra integral representation. Taylor polynomial solution of the linear form is obtained in Section 3. In section 4 we give an algorithm to show the applicability of the method. Finally in Section 5, numerical results of some test problems solved by the proposed method are reported.

2. Newton's Method

Let X and Y be two Banach spaces, $F : X \to Y$ be a Frechet differentiable operator. We are going to introduce Newton's method in solving the equation

$$F(U) = 0. \tag{2.1}$$

The Newton's method reads as follow: choose an initial guess $U^{(0)} \in X$; for $n = 0, 1, ...,$ compute $\Delta^{(n+1)}$ from

$$F^{'}(U^{(n)})\Delta^{(n+1)} = -F(U^{(n)}), \tag{2.2}$$

then $U^{(n+1)} = U^{(n)} + \Delta^{(n+1)}$, and repeat this procedure for better approximation. The following theorem gives evaluation of linear operator $F^{'}$.

Theorem 2.1. *Suppose X and Y be two Banach spaces and $F : X^d \longrightarrow Y^d$ is Frechet differentiable operator at $U^{(0)} = (u_1^{(0)}, u_2^{(0)}, ..., u_d^{(0)})^T$. Then the partial Frechet derivatives of F which shows with*

$$D_i F(U^{(0)}) = \frac{\partial F(U)}{\partial u_i}\Big|_{U=U^{(0)}} = F_{u_i}(U^{(0)}), \tag{2.3}$$

exist and

$$F^{'}(U^{(0)})\Delta U = \sum_{i=1}^{d} F_{u_i}(U^{(0)})\Delta u_i, \quad \Delta U = (\Delta u_1, \Delta u_2, ..., \Delta u_d)^T. \tag{2.4}$$

Conversely, if $F_{u_i}(U^{(0)}), i = 1, ..., d$ exist in a neighborhood of $(u_1^{(0)}, u_2^{(0)}, ..., u_d^{(0)})^T$ and are continuous at $(u_1^{(0)}, u_2^{(0)}, ..., u_d^{(0)})^T$, then F is Frechet differentiable at $(u_1^{(0)}, u_2^{(0)}, ..., u_d^{(0)})^T$, and (2.4) holds.

Proof . See [1] theorem 4.3.14.□

2.1. Application of Newton's Method for the problem (1.4)

Consider the operator

$$\begin{cases} F : (C^1[0,b])^d \longrightarrow (C[0,b])^d, \\ F(U)(t) := U'(t) - f(t, U(t)), \quad U \in (C^1[0,b])^d, t \in [0,b]. \end{cases} \tag{2.5}$$

Then (2.1) is the operator form of (1.4), and the associated Newton's method is (2.2). From theorem 2.1

$$F'(U^{(n)})\Delta U^{(n+1)} = \sum_{i=1}^{d} F_{u_i}(U^{(n)})\Delta u_i^{(n+1)}, \tag{2.6}$$

$$\begin{aligned} F_{u_i}(U)\Delta u &= \lim_{h\to 0} h^{-1}[F(U + h\Delta u e_i) - F(U)] \\ &= (\Delta u)' e_i - \lim_{h\to 0} h^{-1}[f(t, (U + h\Delta u e_i)(t)) - f(t, U(t))] \\ &= (\Delta u)' e_i - \frac{\partial f}{\partial u_i}(t, U(t))\Delta u(t), \end{aligned} \tag{2.7}$$

where e_i is a vector with ith component 1, and the other components are zero. And

$$\frac{\partial f}{\partial u_i}(t, U(t)) = \left(\frac{\partial f_1}{\partial u_i}(t, U(t)), ..., \frac{\partial f_d}{\partial u_i}(t, U(t))\right)^T. \tag{2.8}$$

From equations (2.6) and (2.7), the equation (2.2) reduces to

$$\sum_{i=1}^{d} \left\{ \left(\Delta u_i^{(n+1)}(t)\right)' e_i - \frac{\partial f}{\partial u_i}(t, U^{(n)}(t))\Delta u_i^{(n+1)}(t) \right\} = -F(U^{(n)})(t), \tag{2.9}$$

which is equivalent with

$$\left(\Delta U^{(n+1)}(t)\right)' - f'(t, U^{(n)}(t))\Delta U^{(n+1)}(t) = -F(U^{(n)})(t).$$

Substitution of $\left(\Delta U^{(n+1)}(t)\right)' = \left(U^{(n+1)}(t)\right)' - \left(U^{(n)}(t)\right)'$ forces

$$\left(U^{(n+1)}(t)\right)' - f'(t, U^{(n)}(t))U^{(n+1)}(t) = f(t, U^{(n)}(t)) - f'(t, U^{(n)}(t))U^{(n)}(t), \tag{2.10}$$

where $f'(t, U^{(n)}(t)) = \left[\frac{\partial f_i}{\partial u_j}(t, U(t))|_{U(t)=U^{(n)}(t)}\right]_{d\times d}$ is the Jacobian matrix. This is a linear system of ordinary differential equations, which is reduced from Newton's method. For other related discussion of Newton's method see [5, 10].

2.2. Convergence of Newton's Method

We show that for the operator $F : X \longrightarrow Y$ with the Banach spaces $X := (C^1[0,b])^d, Y := (C[0,b])^d$, all conditions of the following theorem hold for some varies vector functions f. A proof of this theorem can be found in [15].

Theorem 2.2. *(Kantorovich) Suppose that*

1. $F : D(F) \subseteq X \to Y$ *is differentiable on an open convex set* $D(F)$*, and the derivative is Lipschitz continuous, i.e. there exist a positive constant* $Lip > 0$ *such that*

$$\|F'(U) - F'(V)\| \le Lip\|U - V\| \quad \forall U, V \in D(F), \tag{2.11}$$

2. *For some* $U^{(0)} \in D(F), [F'(U^{(0)})]^{-1}$ *exists and is a continuous operator from* Y *to* X*, and such that* $h := \alpha\beta Lip \leq 1/2$ *for some* $\alpha \geq \|[F'(U^{(0)})]^{-1}\|$ *and* $\beta \geq \|[F'(U^{(0)})]^{-1}F(U^{(0)})\|$*. Denote*

$$t^* = \frac{1-(1-2h)^{1/2}}{\alpha Lip}, \quad t^{**} = \frac{1+(1-2h)^{1/2}}{\alpha Lip} \tag{2.12}$$

3. $U^{(1)}$ *is chosen so that* $\overline{B}(U^{(1)}, r) \subseteq D(F)$*, where* $r = t^* - \beta$*.*
Then the equation (2.1) has a solution $U^* \in \overline{B}(U^{(1)}, r)$ *and the solution is unique in* $\overline{B}(U^{(0)}, t^{**}) \cap D(F)$*; the sequence* $\{U^{(n)}\}$ *converges to* U^**, and we have the error estimate*

$$\|U^{(n)} - U^*\| \leq \frac{\left(1-(1-2h)^{1/2}\right)^{2^n}}{2^n \alpha Lip}, \quad n = 0, 1, 2, ... \tag{2.13}$$

The left hand side of (2.10) defines $F'(U^{(n)})U^{(n+1)}$, and hence for all $U, W \in X$ with $U = (u_1, ..., u_d)^T$, $W = (w_1, ..., w_d)^T$ we have

$$F'(U)W = W' - A(U)W, \tag{2.14}$$

where

$$A(U(t)) = \left[\frac{\partial f_i}{\partial u_j}(t, U(t))\right]_{d\times d}, \tag{2.15}$$

where $A(U(t))$ is the Frechet or Jacobian matrix. Here $D(F) = X$, is a Banach space and condition (3) is automatically satisfies. For conditions (1) and (2) we have the following subsections

2.3. The derivative operator is Lipschitz Continuous

In this subsection we prove that for some $b > 0$ there exists $Lip > 0$ such that for $V, \widetilde{V} \in X$ we have

$$\|F'(V) - F'(\widetilde{V})\| \leq Lip\|V - \widetilde{V}\|. \tag{2.16}$$

Or

$$\|A(V) - A(\widetilde{V})\| \leq Lip\|V - \widetilde{V}\|. \tag{2.17}$$

For this aim we give the following theoram

Theorem 2.3. *(Generalized Taylor's Theorem) Let* $A : X \longrightarrow Y$ *be an operator between two Banach spaces, such that* A *is* n *times continuously differentiable in a neighborhood* $b(V, r)$*,* $r > 0$*, of* V*. Then for all* $\widetilde{V}$ *in the interior of* $b(V, r)$

$$\left\| A(V) - A(\widetilde{V}) - \sum_{i=1}^{n-1} \frac{1}{i!} A^{(i)}(V)(V - \widetilde{V})^i \right\| \leq \sup_{W \in l(V,\widetilde{V})} \|A^{(n)}(W)\| \frac{\|V - \widetilde{V}\|^n}{n!}.$$

where $l(V, \widetilde{V})$ *is the line segment between* V *and* $\widetilde{V}$*.*

Proof . See [12], Theorem 5.8. □

The inequality (2.17) is obtain from the above theorem by choosing $n = 1$ and $Lip = \sup_{W\in l(V,\widetilde{V})} \|A'(W)\|$.

2.4. The hypothesis (2) of theorem 2.2 satisfies for some $b > 0$

We have the following theorem about this subsection

Theorem 2.4. *Let X and Y be normed spaces with at least one of them being complete. Assume $L \in \mathcal{L}(X, Y)$ (the set of all continuous linear operators from X to Y) has a bounded inverse $L^{-1} : Y \to X$. Assume $M \in \mathcal{L}(X, Y)$ satisfies $\|M - L\| \leq \frac{1}{\|L^{-1}\|}$. Then $M : X \to Y$ is a bijection, $M^{-1} \in \mathcal{L}(Y, X)$ and $\|M^{-1}\| \leq \frac{\|L^{-1}\|}{1-\|L^{-1}\|\|M-L\|}$.*

Proof . See [1], Theorem 2.3.5. □

Put $U^{(0)} = U_0$, since $U(0) = U_0$ then $F(U^{(0)})(0) = 0$. F is continuous and for a given $\epsilon > 0$ there exist $b > 0$ such that $\|F(U^{(0)})\| < \epsilon$ on $t \in [0, b]$. In theorem 2.4 put $X = (C^1[0, b])^d$, $Y = (C[0, b])^d$, $L(U) = U'$ and $M(U) = F'(U^{(0)})U$, then $\|L\| = \|f\|$ and $\|M - L\| = \|f'(U^{(0)})\|$, which are known for a known f. If $\|f'(U^{(0)})\| \leq \frac{1}{\|L^{-1}\|} \leq \|f\|$, then all hypotheses of theorem 2.4 satisfy and hence $M^{-1} = \left[F'(U^{(0)})\right]^{-1} \in \mathcal{L}(Y, X)$. Thus $\|M^{-1}F'(U^{(0)})\| \leq \|M^{-1}\| \leq \alpha(\epsilon) \to 0$ as $\epsilon \to 0$, and hence we can choose $\epsilon > 0$ and associated $b > 0$ such that $h = \alpha(\epsilon)\beta(\epsilon)Lip \leq \frac{1}{2}$. For these problems the hypothesis of (2) of theorem 2.2 is true.

3. The Taylor polynomial solutions Technique Applied to One Step of Newton's Method

The linear differential equations (2.10) will be solved by the method is described in [14]. We are going to explain this technique for One Step of Newton's Method. For this purpose put $P_1(t) = I_d$ (the $d \times d$ identity matrix), $P_0(t) = -f'(t, U^{(n)}(t))$, $y(t) = U^{(n+1)}(t)$ and $r(t) = f(t, U^{(n)}(t)) - f'(t, U^{(n)}(t))U^{(n)}(t)$, then (2.10) reduces to

$$P_0(t)y(t) + P_1(t)y'(t) = r(t). \tag{3.1}$$

which is in the form (8) of [14]. Suppose we are going to solve (3.1) on a short interval $[a, b]$. For this aim we represent the solution by a truncated Taylor series

$$y_i(t) = \sum_{j=0}^{N} \frac{y_i^{(j)}(c)}{j!}(t - c)^j, i = 1, ..., d, a \leq c \leq b. \tag{3.2}$$

Where $N \geq 1$ is any positive integer and $y_i^{(j)}(c)$ are the Taylor coefficients to be determined. Derivation of (3.2) yields

$$y_i'(t) = \sum_{j=1}^{N} \frac{y_i^{(j)}(c)}{(j-1)!}(t - c)^{j-1}, i = 1, ..., d. \tag{3.3}$$

Functions defined by equations (3.2)-(3.3) can be written in the matrix form

$$y_i^{(m)}(t) = T(t)M_mA_i, i = 1, ..., d, m = 0, 1, \tag{3.4}$$

where

$$T(t) = \left(1, (t - c), (t - c)^2, ..., (t - c)^N\right),$$

$$A_i = \left(y_i(c), y_i'(c), y_i''(c), ..., y_i^N(c)\right)^T,$$

$$M_0 = \begin{bmatrix} \frac{1}{0!} & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{1!} & 0 & \dots & 0 \\ 0 & 0 & \frac{1}{2!} & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & \frac{1}{N!} \end{bmatrix}, M_1 = \begin{bmatrix} 0 & \frac{1}{0!} & 0 & 0 & \dots & 0 \\ 0 & 0 & \frac{1}{1!} & 0 & \dots & 0 \\ 0 & 0 & 0 & \frac{1}{2!} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \frac{1}{(N-1)!} \\ 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix}.$$

From (3.4) we can write

$$\begin{aligned} y^{(i)}(t) &= \begin{bmatrix} y_1^{(i)}(t) \\ y_2^{(i)}(t) \\ \vdots \\ y_d^{(i)}(t) \end{bmatrix} = \begin{bmatrix} T(t)M_iA_1 \\ T(t)M_iA_2 \\ \vdots \\ T(t)M_iA_d \end{bmatrix} \\ &= \begin{bmatrix} T(t) & 0 & 0 & 0 \\ 0 & T(t) & 0 & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & T(t) \end{bmatrix} \begin{bmatrix} M_i & 0 & 0 & 0 \\ 0 & M_i & 0 & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & M_i \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_d \end{bmatrix} \\ &= T^*(t)M_i^*A \end{aligned} \tag{3.5}$$

where

$$T^*(t) = \begin{bmatrix} T(t) & 0 & 0 & 0 \\ 0 & T(t) & 0 & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & T(t) \end{bmatrix}_{d\times d \ Blocks} \in \mathbb{R}^{d\times(N+1)d},$$

$$M_i^* = \begin{bmatrix} M_i & 0 & 0 & 0 \\ 0 & M_i & 0 & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & M_i \end{bmatrix}_{d\times d \ Blocks} \in \mathbb{R}^{(N+1)d\times(N+1)d},$$

$$A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_d \end{bmatrix}_{d\times 1 \ Blocks} \in \mathbb{R}^{(N+1)d\times 1}.$$

Now we evaluate the Taylor coefficients by introducing of the Taylor collocation points

$$t_j = a + \frac{b-a}{N}j, \qquad j = 0, 1, ..., N. \tag{3.6}$$

Substitution of the collocation points (3.6) in to the matrix equation (3.1) forces

$$P_0Y^{(0)} + P_1Y^{(1)} = R, \tag{3.7}$$

where

$$R = \begin{bmatrix} r(t_0) \\ r(t_1) \\ \vdots \\ r(t_N) \end{bmatrix}_{(N+1)\times 1 \ Blocks} \in \mathbb{R}^{(N+1)d\times 1},$$

$$P_i = \begin{bmatrix} P_i(t_0) & 0 & 0 & 0 \\ 0 & P_i(t_1) & 0 & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & P_i(t_N) \end{bmatrix}_{(N+1)\times(N+1)\ Blocks} \in \mathbb{R}^{(N+1)d\times(N+1)d}, \quad i=0,1,$$

$$Y^{(i)} = \begin{bmatrix} y^{(i)}(t_0) \\ y^{(i)}(t_1) \\ \vdots \\ y^{(i)}(t_N) \end{bmatrix} = \begin{bmatrix} T^*(t_0)M_i^*A \\ T^*(t_1)M_i^*A \\ \vdots \\ T^*(t_N)M_i^*A \end{bmatrix} = TM_i^*A, \quad i=0,1, \tag{3.8}$$

where

$$T = \begin{bmatrix} T^*(t_0) & T^*(t_1) & \dots & T^*(t_N) \end{bmatrix}_{1\times(N+1)\ Blocks} \in \mathbb{R}^{(N+1)d\times(N+1)d}.$$

Substitution of (3.8) in (3.7) forces

$$P_0TM_0^*A + P_1TM_1^*A = R. \tag{3.9}$$

Define $W = [w_{ij}]_{(N+1)d\times(N+1)d} := P_0TM_0^* + P_1TM_1^*$, then we must solve the following linear system of algebric equations

$$WA = R. \tag{3.10}$$

This system gives a general solution for $y(t)$. For a particular solution that satisfies the initial condition

$$y(a) = \lambda, \tag{3.11}$$

where $\lambda = (\lambda_1, ..., \lambda_d)^T$ is a known initial vector, we do as the following procedure.
From (3.5) we have

$$T^*(a)M_0^*A = \lambda. \tag{3.12}$$

If we define $V := T^*(a)M_0^*A$, then the fundamental matrix form of initial conditions is

$$VA = \lambda. \tag{3.13}$$

Replace the rows of the matrix V and λ, by the last rows of the matrix W and R, respectively and obtain

$$\widetilde{W}A = \widetilde{R}, \tag{3.14}$$

where

$$\widetilde{W} = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1,(N+1)d} \\ w_{21} & w_{22} & \dots & w_{2,(N+1)d} \\ \vdots & \vdots & \dots & \vdots \\ w_{Nd,1} & w_{Nd,2} & \dots & w_{Nd,(N+1)d} \\ v_{11} & v_{12} & \dots & v_{1,(N+1)d} \\ v_{21} & v_{22} & \dots & v_{2,(N+1)d} \\ \vdots & \vdots & \dots & \vdots \\ v_{d,1} & v_{d,2} & \dots & w_{d,(N+1)d} \end{bmatrix}, \widetilde{R} = \begin{bmatrix} r(t_0) \\ r(t_1) \\ \vdots \\ r(t_{N-1}) \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_d \end{bmatrix}.$$

By solving the linear algebraic equation (3.14) we obtain the particular solution of the initial value problem

$$\begin{cases} P_0(t)y(t) + P_1(t)y^{'}(t) = r(t), \\ y(a) = \lambda. \end{cases} \tag{3.15}$$

4. Algorithm of Newton-Taylor polynomial solutions

In this section we give an algorithm for the initial value problem (3.15) on $[0, bN_i] = \cup_{i=0}^{N_i-1}[bi, b(i+1)]$, where $N_i \in \mathbb{N}$ and $b > 0$ are the number and length of partial intervals, respectively.
Note that in this algorithm we extend the method of described in [14] from two points of view. First we apply the method for nonlinear problems and second we extend the method for union of short intervals, which are the advantages of the algorithm.

Step1 Input integers d, N, N_i and N_n (number of Newton's iteration), real $b > 0$, initial vector $\lambda = \left(u_1^{(0)}, ..., u_d^{(0)}\right)^T$ and vector valued function $f(t, U(t))$;
Evaluate M_0, M_1, M_0^*, M_1^* and P_1 as mentioned in section 3;
Set $a = 0$, $length = b-a$, $U^{(0)}(t) = \left(\begin{cases} u_1^{(0)} & t \le b \\ 0 & otherwise \end{cases}, ..., \begin{cases} u_d^{(0)} & t \le b \\ 0 & otherwise \end{cases}\right)^T$ and $\widetilde{U}(t) = U^{(0)}(t)$;
Step2
step 2.1 Set $i_1 = 1$; (interval loop)
step2.2 If $i_1 > N_i$, go to Step3;
Set $c = \frac{a+b}{2}$;
$t_j = a + \frac{b-a}{N} j \quad j = 0, 1, ..., N$;
Evaluate matrices $T(t), T^*(t), T$ in the current interval as introduced in Section 3;
step2.3 Set $i_2 = 1$; (Newton's loop)
step2.4 If $i_2 > N_n$, go to step2.5;
Evaluate matrix P_0 in the current interval as introduced in Section 3;
Set W=$P_0TM_0^* + P_1TM_1^*$;
$V = T^*(a)M_0^*$;
Remove d rows of W and replace it by V, then put the new matrix in $\widetilde{W}$;
Set $r(t) = f(t, U^{(0)}(t)) - f'(t, U^{(0)}(t))U^{(0)}(t)$;

$$\widetilde{R} = \begin{bmatrix} r(t_0) \\ r(t_1) \\ \vdots \\ r(t_{N-1}) \\ \lambda \end{bmatrix};$$

Solve the linear system $\widetilde{W}A = \widetilde{R}$ and obtain A;

$$\text{Set } \widetilde{U}(t) = \left(\begin{cases} \widetilde{u}_1(t) & t \le a \\ (T^*(a)M_0^*A)_1 & a < t \le b \\ 0 & otherwise \end{cases}, ..., \begin{cases} \widetilde{u}_d(t) & t \le a \\ (T^*(a)M_0^*A)_d & a < t \le b \\ 0 & otherwise \end{cases}\right)^T;$$

$U^{(0)}(t) = \widetilde{U}(t)$;
$i_2 = i_2 + 1$ and go to step 2.4;
step2.5 Set $\lambda = \widetilde{U}(b)$;
a=b;
b=length+b;
$i_1 = i_1 + 1$ and go to step 2.2;
Step3 Plot $\widetilde{U}(t)$ on $[0, bN_i]$ as the approximated solution by the proposed method, and end the algorithm.

5. Numerical Examples and Discussion

Example 5.1. *Consider the initial value problem* (1.1) *with*

$$f(t,U(t)) = \begin{bmatrix} u_2(t) - u_1^2(t) \\ 2u_2(t)u_1(t) - 2u_1^3(t) \end{bmatrix}, U_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

This system has the exact solution $U(t) = \begin{bmatrix} t \\ t^2+1 \end{bmatrix}$. *We solve this system by two methods. In Table 1 we give the Newton-Taylor polynomial solutions method with* $b = 0.2$*, which is the length of the partial intervals,* $N = 3$*,* $N_n = 5$*,* $N_i = 25$ *and obtain the solution on* $[0, 5]$*. In Table 2, we give the Runge-Kutta fourth-order method with step length* $h = 0.1$ *on* $[0, 5]$*. In Table 1 and 2, columns 2,3 shows absolute errors of* $\widetilde{u}_1$ *and* $\widetilde{u}_2$ *at* $t_i = 0.5i, i = 0, 1, ..., 10$*, and columns 4,5 show relative errors of* $\widetilde{u}_1$ *and* $\widetilde{u}_2$ *at* $t_i = 0.5i, i = 0, 1, ..., 10$*, by the proposed and Runge-Kutta fourth-order methods, respectively.* u_1, u_2 *are exact solutions and* $\widetilde{u}_1, \widetilde{u}_2$ *are the approximated solutions. In all Tables of this paper, neg means negligible. As Table 1 shows the accuracy of the proposed method has a superconvergence result, whereas the famous Runge-Kutta fourth-order method has an ordinary convergence accuracy.*

Table 1: Absolute and relative errors of $\widetilde{u}_1$ and $\widetilde{u}_2$ for Example 5.1. by the Newton-Taylor polynomial solutions method

i	$\lvert u_1(t_i) - \widetilde{u}_1(t_i)\rvert$	$\lvert u_2(t_i) - \widetilde{u}_2(t_i)\rvert$	$\left\lvert\frac{u_1(t_i)-\widetilde{u}_1(t_i)}{u_1(t_i)}\right\rvert$	$\left\lvert\frac{u_2(t_i)-\widetilde{u}_2(t_i)}{u_2(t_i)}\right\rvert$
0	*neg*	*neg*		
1	1.1×10^{-16}	*neg*	2.2×10^{-16}	*neg*
2	6.7×10^{-16}	*neg*	6.7×10^{-16}	*neg*
3	2.2×10^{-16}	8.9×10^{-16}	1.5×10^{-16}	2.7×10^{-16}
4	*neg*	*neg*	*neg*	*neg*
5	*neg*	3.6×10^{-15}	*neg*	4.9×10^{-16}
6	4.0×10^{-15}	3.0×10^{-14}	1.3×10^{-15}	3.0×10^{-16}
7	7.6×10^{-15}	8.2×10^{-14}	2.2×10^{-15}	6.2×10^{-15}
8	2.1×10^{-14}	2.1×10^{-13}	5.3×10^{-15}	1.3×10^{-14}
9	4.3×10^{-14}	4.2×10^{-13}	9.5×10^{-15}	2.0×10^{-14}
10	4.1×10^{-14}	5.4×10^{-13}	8.2×10^{-15}	2.1×10^{-14}

Example 5.2. *Consider the initial value problem* (1.1) *with*

$$f(t,U(t)) = \begin{bmatrix} u_1^2(t) - u_2(t) - 2u_1(t) + 2 \\ 2u_1^2(t) - 2u_1(t) - 2u_2(t) \\ 3u_1(t)u_2(t) - 3u_3(t) \end{bmatrix}, U_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

This system has the exact solution $U(t) = \begin{bmatrix} 1+t \\ t^2 \\ t^3 \end{bmatrix}$. *Again we solve this system by two methods. In Tables 3 and 4 we give the Newton-Taylor polynomial solutions method with* $b = 0.2$*, which is the length of the partial intervals,* $N = 3$*,* $N_n = 5$*,* $N_i = 25$ *and obtain the solution on* $[0, 5]$*. In Tables 5 and 6, we give the Runge-Kutta fourth-order method with step length* $h = 0.05$ *on* $[0, 5]$*. In*

Table 2: Absolute and relative errors of $\widetilde{u}_1$ and $\widetilde{u}_2$ for Example 5.1. by the Runge-Kutta fourth-order method

i	$\lvert u_1(t_i) - \widetilde{u}_1(t_i)\rvert$	$\lvert u_2(t_i) - \widetilde{u}_2(t_i)\rvert$	$\left\lvert\frac{u_1(t_i)-\widetilde{u}_1(t_i)}{u_1(t_i)}\right\rvert$	$\left\lvert\frac{u_2(t_i)-\widetilde{u}_2(t_i)}{u_2(t_i)}\right\rvert$
0	neg	neg		
1	4.2×10^{-6}	4.3×10^{-6}	8.4×10^{-6}	3.4×10^{-6}
2	8.5×10^{-6}	1.7×10^{-5}	8.5×10^{-6}	8.4×10^{-6}
3	1.3×10^{-5}	3.9×10^{-5}	8.5×10^{-6}	1.2×10^{-5}
4	1.7×10^{-5}	6.9×10^{-5}	8.6×10^{-6}	1.4×10^{-5}
5	2.2×10^{-5}	1.1×10^{-4}	8.6×10^{-6}	1.5×10^{-5}
6	2.6×10^{-5}	1.6×10^{-4}	8.7×10^{-6}	1.6×10^{-5}
7	3.1×10^{-5}	2.1×10^{-4}	8.7×10^{-6}	1.6×10^{-5}
8	3.5×10^{-5}	2.8×10^{-4}	8.8×10^{-6}	1.7×10^{-5}
9	4.0×10^{-5}	3.6×10^{-4}	8.8×10^{-6}	1.7×10^{-5}
10	4.4×10^{-5}	4.5×10^{-4}	8.9×10^{-6}	1.7×10^{-5}

Tables 3 and 5, columns 2,3,4 show absolute errors of $\widetilde{u}_j$, $j = 1, 2, 3$ *at* $t_i = 0.5i, i = 0, 1, ..., 10,$ *and in Tables 4 and 6 columns 2,3,4 show relative errors of* $\widetilde{u}_j$, $j = 1, 2, 3$ *at* $t_i = 0.5i, i = 0, 1, ..., 10,$ *by the proposed and Runge-Kutta fourth-order methods, respectively.* u_1, u_2, u_3 *are exact solutions and* $\widetilde{u}_1, \widetilde{u}_2, \widetilde{u}_3$ *are the approximated solutions. As these tables show the accuracy of the Runge-Kutta fourth-order method is not bad, but the accuracy of the proposed method is very good.*

Table 3: Absolute errors of $\widetilde{u}_1, \widetilde{u}_2$ and $\widetilde{u}_3$ for Example 5.2. by the Newton-Taylor polynomial solutions method

i	$\lvert u_1(t_i) - \widetilde{u}_1(t_i)\rvert$	$\lvert u_2(t_i) - \widetilde{u}_2(t_i)\rvert$	$\lvert u_3(t_i) - \widetilde{u}_3(t_i)\rvert$
0	neg	neg	neg
1	2.2×10^{-16}	2.2×10^{-16}	5.6×10^{-17}
2	4.4×10^{-16}	4.4×10^{-16}	2.2×10^{-16}
3	1.8×10^{-15}	8.9×10^{-16}	5.8×10^{-15}
4	3.1×10^{-15}	1.1×10^{-14}	2.8×10^{-14}
5	3.1×10^{-15}	1.6×10^{-14}	5.7×10^{-14}
6	1.8×10^{-15}	1.1×10^{-14}	7.5×10^{-14}
7	2.6×10^{-14}	4.6×10^{-14}	1.7×10^{-13}
8	3.6×10^{-13}	8.1×10^{-13}	3.2×10^{-12}
9	7.8×10^{-12}	1.8×10^{-11}	7.8×10^{-11}
10	2.8×10^{-10}	6.5×10^{-10}	3.0×10^{-9}

Acknowledgments

The author is thankful to the referees for their valuable suggestions to improve this paper.

Table 4: Relative errors of $\widetilde{u}_1, \widetilde{u}_2$ and $\widetilde{u}_3$ for Example 5.2. by the Newton-Taylor polynomial solutions method

i	$\left\|\frac{u_1(t_i)-\widetilde{u}_1(t_i)}{u_1(t_i)}\right\|$	$\left\|\frac{u_2(t_i)-\widetilde{u}_2(t_i)}{u_2(t_i)}\right\|$	$\left\|\frac{u_3(t_i)-\widetilde{u}_3(t_i)}{u_3(t_i)}\right\|$
1	1.5×10^{-16}	8.9×10^{-16}	4.4×10^{-16}
2	2.2×10^{-16}	4.4×10^{-16}	2.2×10^{-16}
3	7.1×10^{-16}	3.9×10^{-16}	1.7×10^{-15}
4	1.0×10^{-15}	2.8×10^{-15}	3.6×10^{-15}
5	8.9×10^{-16}	2.6×10^{-15}	3.6×10^{-15}
6	4.4×10^{-16}	1.2×10^{-15}	2.8×10^{-15}
7	5.7×10^{-15}	3.8×10^{-15}	4.0×10^{-15}
8	7.1×10^{-14}	5.1×10^{-14}	5.0×10^{-14}
9	1.4×10^{-12}	8.9×10^{-13}	8.6×10^{-13}
10	4.7×10^{-11}	2.6×10^{-11}	2.4×10^{-11}

Table 5: Absolute errors of $\widetilde{u}_1, \widetilde{u}_2$ and $\widetilde{u}_3$ for Example 5.2. by the Runge-Kutta fourth-order method

i	$\|u_1(t_i) - \widetilde{u}_1(t_i)\|$	$\|u_2(t_i) - \widetilde{u}_2(t_i)\|$	$\|u_3(t_i) - \widetilde{u}_3(t_i)\|$
0	*neg*	*neg*	*neg*
1	5.2×10^{-8}	3.2×10^{-7}	1.6×10^{-6}
2	1.3×10^{-7}	6.2×10^{-7}	1.8×10^{-6}
3	2.1×10^{-7}	8.7×10^{-7}	1.8×10^{-6}
4	2.4×10^{-7}	9.6×10^{-7}	1.5×10^{-6}
5	1.1×10^{-7}	5.7×10^{-7}	1.6×10^{-7}
6	9.7×10^{-7}	2.3×10^{-6}	1.1×10^{-5}
7	1.0×10^{-5}	2.6×10^{-5}	9.9×10^{-5}
8	1.4×10^{-4}	3.3×10^{-4}	1.3×10^{-3}
9	3.0×10^{-3}	6.9×10^{-3}	3.0×10^{-2}
10	1.1×10^{-1}	2.5×10^{-1}	1.2

Table 6: Relative errors of $\widetilde{u}_1, \widetilde{u}_2$ and $\widetilde{u}_3$ for Example 5.2. by the Runge-Kutta fourth-order method

i	$\left\|\frac{u_1(t_i)-\widetilde{u}_1(t_i)}{u_1(t_i)}\right\|$	$\left\|\frac{u_2(t_i)-\widetilde{u}_2(t_i)}{u_2(t_i)}\right\|$	$\left\|\frac{u_3(t_i)-\widetilde{u}_3(t_i)}{u_3(t_i)}\right\|$
1	3.5×10^{-8}	1.3×10^{-6}	1.3×10^{-5}
2	6.7×10^{-8}	6.2×10^{-7}	1.8×10^{-6}
3	8.3×10^{-8}	3.9×10^{-7}	5.2×10^{-7}
4	8.0×10^{-8}	2.4×10^{-7}	1.9×10^{-7}
5	3.0×10^{-8}	9.2×10^{-8}	1.0×10^{-8}
6	2.4×10^{-7}	2.5×10^{-7}	3.9×10^{-7}
7	52.3×10^{-6}	2.1×10^{-6}	2.3×10^{-6}
8	2.8×10^{-5}	2.1×10^{-5}	2.1×10^{-5}
9	5.4×10^{-4}	3.4×10^{-4}	3.3×10^{-4}
10	1.9×10^{-3}	1.0×10^{-2}	9.5×10^{-3}

References

[1] K. Atkinson and W. Han, *Theoritical Numerical Analysis*, Springer, New York, 2001.

[2] B. Babayar-Razlighi and B. Soltanalizadeh, *Numerical solution of nonlinear singular Volterra integral system by the Newton-Product integration method*, Math. Comput. Model. 58 (2013) 1696–1703.

[3] B. Babayar-Razlighi, *Extrapolation method for Numerical solution of a model for endemic infectious diseases*, Mathematical Researches 5 (1) (2019) 29–38 (In Persian).

[4] B. Babayar-Razlighi, K. Ivaz and M. R. Mokhtarzadeh, *Newton-Product Integration for a Stefan Problem with Kinetics*, J. Sci. I. R. of Iran 22(1) (2011) 51–61.

[5] M. Berger, *Nonlinearity and Functional Analysis*, Academic Press, New York, 1997.

[6] H. Brunner, *Implicitly linear collocation methods for nonlinear Volterra equations*, Appl. Numer. Math. 9 (1992) 235-247.

[7] R.L. Burden and J.D. Faires, *Numerical Methods*, 3st ed., Thomson/Brooks/Cole, 2003.

[8] E.A. Coddington and N. Levinson, *Theory of Ordinary Differential Equations*, McGraw Hill, New York, 1955.

[9] L.M. Delves and J.L. Mohamed, *Computational methods for integral equations*, Cambridge University Press, 1985.

[10] L. Kantorovich, and G. Akilov, *Nonlinearity and Functional Analysis in Normed Spaces. 2nd ed*, Pergamon Press, New York, 1982.

[11] P. Linz, *Analytical and numerical methods for Volterra equations*, SIAM Philadelphia, 1985.

[12] P. Linz, *Theoretical numerical Analysis*, John Wiley and sons Inc., 1979.

[13] K. Maleknejad and M. Hadizadeh, *Numerical Study of nonlinear Volterra integro-differential equations by Adomian's method*, J. Sci. I.R. Iran 9 (1998) 51–58.

[14] M. Sezer, A. Karamete and M. Gulsu, *Taylor polynomial solutions of systems of linear differential equations with variable coefficients*, Int. J. Comput. Math. 82(6) (2005) 755-764.

[15] E. Zeidler, *Nonlinear Functional analysis and its applications. I: Fixed point Theorems*, Springer-Verlog, New York, 1985.