On the Numerical Solutions Based on a Novel Hybrid Method for Some VNDEs Problems

Stipan Podobnic[1,*], Barbara Charchekhandra[2]

[1]Department of Mathematics, University of Rijeka, City of Rijeka, Croatia

[2]Jadavpur University, Department Of Mathematics, Kolkata, India

Emails: Stipanpod133@Uniri.hr; Charchekhandrabar32@yahoo.com

Abstract

This paper is devoted to find the solution of the Vanishing Neutral Differential Equations (VNDEs), where we review the ARCHI code for solving neutral differential equations, and then an improvement of this code will be presented in this paper to solve problems of VNDEs. This improvement is done by suggesting a new hybrid method of special interpolants with iteration procedure of RK method. We will analyze the convergence of the suggested new method by proving the following criteria

$$\beta_n(t) \leq TH\left(c_1 + \frac{c_2 L_3}{1 - L_3}\right) exp\left(\frac{LT}{1 - L_3}\right)$$

Where the solution is convergent $\boldsymbol{\beta_n(t) \to 0}$, we have $\boldsymbol{H \to 0}$ and when and the derivative of the solution is also convergent for VNDEs.

Keywords: Hybrid method; Numerical Solution; Convergence analysis; VDNE problem

1. Introduction

Many physical systems have the characteristic of delayed response in relation to the incoming conditions. Accordingly, the ratio at which processes appear or occur depends not only on the current calculated state of the system, but also on previous States. The mathematical models of such equations are represented by time-delayed differential equations. Numerical solutions of time-delayed differential equations of the delayed (vanishing) type have been discussed by many scientists recently, among them: Neves, in his research in 1981[9], he prepared two error control mechanisms that address discontinuity, he monitored and monitored the difference between the delay value calculated by extrapolation completion from the previous step and the delay value calculated by local interpolation [3,4]. The time-delayed of the neutral differential equations (NDEs) receive little attention, although there are programs (codes) such as (ARCHI) and (DRKLAG) [7,8] have been designed for this type of equations, this happens because the time-delayed differential equations of the neutral type (NDEs) are difficult to analyze mathematically because the spread of discontinuities of derivatives is not smoothing and that the limits for the derivative of the delay value must be approximated .[5,6] derivative delay term two methods of implicit one step methods with optional rank P have been discussed by [10] Jackiewict to solve problems Time-delayed differential equations of the neutral type NDEs are built on the basis of Picard iteration. Jackiewicz has also published a series of papers investigating solutions to time-delayed differential equations of neutral type (NDEs) using numerical methods. Scientists Hayashi in 1996 [4] and Champine in 2000 [7] discussed the development and analysis of numerical methods for solving problems of time-delayed differential equations of the two types retarded differential equations (RDEs), reciprocal (neutral) (NDEs) and vanishing delay.

DOI: https://doi.org/10.54216/GJMSA.0110101

Received: October 20, 2023 Revised: February 25, 2024 Accepted: June 20, 2024

In this research, the hybrid method between special inclusion and the iterative method of solving VNDEs problems was proposed. The convergence of this hybrid method was also analyzed. Then provide a description of the ARCHI developer program that we have developed and which has been applied to various issues of the VNDE type.

2. Inclusion based on two time periods

The Runge – Cota formula with the delay gradient formula can be applied to time-delayed differential equations of the neutral type (NDEs), and at each step, we obtain: [4]

$$y_n(t_n + \theta h_n) = y_{n-1}(t_n) + h_n \sum_{i=1}^{\bar{s}+1} \bar{b_\iota}(\theta) K_i$$

$$t_{n,i} = t_n + c_i h_n$$

$$K_i = f(t_{n,i}, Y_i, y(t_{n,i} - \tau_1(t_{n,i}, Y_i)), y(t_{n,i} - \tau_2(t_{n,i}, Y_i)), \dots, y(t_{n,i} - \tau_\mu(t_{n,i}, Y_i)),$$
$$y'(t_{n,i}, \tau_{\mu+1}(t_{n,i}, Y_i)), \dots, y'(t_{n,i}, -\tau_{\mu+w}(t_{n,i}, Y_i)))$$

$$Y_i = y_{n-1}(t_n) + h_n \sum_{j=1}^{i-1} a_{ij} K_j$$

For values of $i = 1,2, \dots, \bar{s} + 1$ since $y_{-1}(t_0) = \phi t_0$ and that y(t) represent approximations to the solution with:

$y(t) = \phi(t)$ for values $t \leq t_0$

$y(t) = y_n(t)$ for values $t \in [t_n, t_{n+1}]$

Where c_i, b_i, a_{ij} represent the coefficients of the RK method, represented by the following figure:

c_1	0					
c_2	$a_{2,1}$	0				
M	M	M				
c_s	a_{s1}	a_{s2}	…	$a_{s,s1}$	0	
M	M					
$c_{\bar{s}+1}$	$c_{\bar{s}+1,1}$	$c_{\bar{s}+1,2}$	…	…	$c_{\bar{s}+1,\bar{s}}$	0
	$\bar{b}_1(\theta)$	$\bar{b}_2(\theta)$	…	…	$\bar{b}_{\bar{s}}(\theta)$	$\bar{b}_{\bar{s}+1}(\theta)$

Figure 1. The coefficients of the RK method

This standard formula described above with the formula for inclusion can be used effectively in solving non-vanishing delay problems, but it is not directly suitable for vanishing delay problems, the reason is that the inclusion can be only when all phase values on the current step are available.

Perhaps the delay time of the current step and the values of the delayed solution needed to calculate the calculated current phase values are not available. The inclusion will be extended from the previous step to the calculated one, but this method is not always convenient in solving problems with non - vanishing lag time due to the local-truncation error, that, the cutting error on the interval $[t_{n-1}, t_{n-1} + 2\, h_{n-1}]$ compared to that on the interval $[t_{n-1}, t_{n-1} + \, h_{n-1}]$, and this can be illustrated by the sixth order, in which the local error on the interval $[t_{n-1}, t_{n-1} + 2\, h_{n-1}]$ is greater than that on the interval $[t_{n-1}, t_{n-1} + \, h_{n-1}]$, although it is not easy to expand or develop the inclusion on the next step in its entirety. It is possible to imagine an expansion or development of the inclusion for just one-step. Our basic idea is to prepare a special interpolation based on all the available information from the previous step and the early factors of the calculated step.

DOI: https://doi.org/10.54216/GJMSA.0110101
Received: October 20, 2023 Revised: February 25, 2024 Accepted: June 20, 2024

When calculating the value of the phase $f\left(t_n + c_{i+1}h_n,\ Y_i,\ y'(.)\right)$. In which the delay is located in the calculated step, we will make an inclusion on the period $[t_{n-1}, t_n + c_i h_n]$based on the last step and stages i of the calculated step and then expand this inclusion on the period $[t_{n-1}, t_n + c_{i+1}h_n]$ to be able to calculate any delay value related to the values of this stage. Thus, we will have provided an inclusion of each stage in the current step when necessary. And note that here we make the assumption that:

$c_2 < c_3 <\ldots< c_m < 1$ and that c_2 checks $c_2 h_n < h_{n-1}$ is negative or positive.

This condition is met for most uses of the commonly used range-Cota formulas and for most integral step size control strategies. If the delay occurs in the period $[t_n, t_n + c_2 h_n]$ and there is no new information available, the calculation of the second step will be only by expanding the interpolation from the previous step. In this case, we can use the standard Runge-Kutta Formula [1].

After the second period, when the delay time gets the period $[t_n + c_2 h_n, t_n + h_n]$in the calculation of the (i+2)th phase, we construct an inclusion in the following form:

$$u_i(t_{n-1} + \theta h_{n-1}) = y_{n-1} + h_{n-1}\sum_{j=1}^{s} b_{i,j}(\theta)\tilde{Y}_j + h_{n-1}\sum_{j=1}^{i+1} b_{i,s+j}(\theta)Y_j\,; \quad 0 \le \theta \le 1 + rci + 1$$

If $\tilde{Y}_j, Y_j$ represent the phases associated with the last and current step, respectively, and $r = h_n/h_{n-1}$

To simplify the issue, we do not use additional stages $\overline{S} - S$ in the construction of a special gradient (the rank in it cannot be increased during the use of additional stages).

We denote the inclusion on the interval $[t_{n-1}, t_n + c_2 h_n]$ by the symbol u_1 for the inclusion on the interval $[t_{n-1}, t_n + c_{i+1}h_n]$ for values $i = 1,2,\ldots,s_1$ since s_1 represents the largest number of necessary inclusions.

Thus, we have:

$$Y_i = y_n + h_n\sum_{j=1}^{i-1} a_{i,j}\, f(t_n + h_n c_i, Y_j, z'(t_n + h_n c_i - \tau(t_n + h_n c_i, Y_i)))$$

$$= y_{n-1} + h_{n-1}\sum_{j=1}^{s} b_i f\left(t_{n-1} + c_j h_{n-1}, \tilde{Y}_j, z'\left(t_{n-1} + c_j h_{n-1} - \tau\left(t_{n-1} + c_j h_{n-1}, \tilde{Y}_i\right)\right)\right) +$$

$$h_{n-1}\sum_{j=1}^{i-1} a_{ij}\frac{h_n}{h_{n-1}}\, f(t_{n-1} + h_{n-1}(1 + \frac{h_n}{h_{n-1}}c_i), Y_j,$$

$$z'(t_{n-1} + h_{n-1}(1 + \frac{h_n}{h_{n-1}}c_i) - \tau\left(t_{n-1} + h_{n-1}(1 + \frac{h_n}{h_{n-1}}c_i)\right), Y_j\Big)$$

The following table can represent such a type of inclusion:

c_1	0									
c_2	$a_{2,1}$	0								
c_3	$a_{3,1}$	$a_{3,2}$	0							
M										
c_s	$a_{s,1}$	$a_{s,2}$	…	$a_{s,s-1}$	0					
1	$b_{1,1}$	$b_{2,2}$	…	$b_{s-1,s-1}$	$b_{s,s}$	0				
$1+ r\,c_2$	b_1	b_2	…	b_{s-1}	b_s	$ra_{2,1}$	0			
$1+ r\,c_3$	b_1	b_2	…	b_{s-1}	b_s	$ra_{3,1}$	$ra_{3,2}$	0		
M	M									
$1 + r\,c_{i+1}$	b_1	b_2	…	b_{s-1}	b_s	$ra_{i+1,1}$	$ra_{i+1,2}$	…	ra_{i+1},i	0
	$b_{i,1}(\theta)$	$b_{i,2}(\theta)$	…	$b_{i,s-1}(\theta)$	$b_{i,s}(\theta)$	$b_{i,s+1}(\theta)$	$b_{i,s+2}(\theta)$	…	$b_{i,s+i}(\theta)$	$b_{i,s+i+1}(\theta)$

Figure 2. A type of inclusion

DOI: https://doi.org/10.54216/GJMSA.0110101
Received: October 20, 2023 Revised: February 25, 2024 Accepted: June 20, 2024

The coefficients $b_{i,j}(\theta) = \sum_{\ell=1}^{p} b_{i,j,\ell}(\theta)^{\ell}$represent polynomials.

3. The hybrid method between the special inclusion and the iterative method of the RK method for solving problems (VNDEs)

The hybrid method has been adapted for use in solving VNDEs problems by forming initial approximations or special interpolation based on all available information from the previous step and the first stages of the current calculated step. This inclusion is based on two times (Two-time step intervals) built in Paragraph (2). This new inclusion is then used together with the iterative methods of the Rang-Kota method.

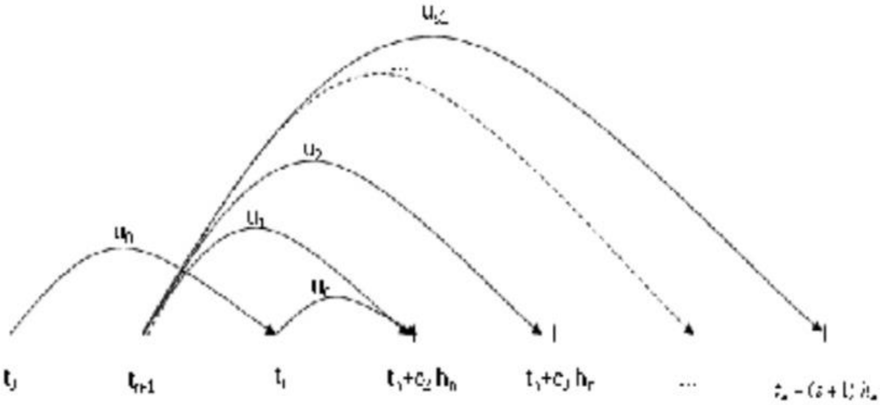

Figure 3. The new inclusion (The hybrid method)

Since u_0 represents the inclusion on the interval $[t_0, t_n]$ and u_i represents the inclusion on the interval $[t_{n-1}, t_n + c_{i+1}h_n]$ for all values $i = 1,2,\ldots,s_1$, since s_1 represents the largest number of special inclusions for each step and if the delay is left within the period $[t_n, t_n + c_2h_n]]$ then we will use u_0 as a special insertion, and this will be explained in detail by algorithms (1) and (2).

Algorithm (1) Initial Approximations or Special Interpolation:

Step (1): Make: $\boldsymbol{K_1^{(0)} = f(t_n, y_{n-1}(t_n), Z'_0(t_n - \tau(t_n, y_{n-1}(t_n)))}$

Step (2): Perform steps (2) and (3) for the values $i = 2, \bar{s} + 1$

$$Y_i^0 = y_{n-1}(t_n) + h_n \sum_{j=1}^{i-1} a_{ij}\, K_j^{(0)}$$

Step (3): if $t_n + c_ih_n - \tau(t_n + c_ih_n, Y_i^{(0)}) \leq t_n + c_2h_n$

Then:

$$K_i^{(0)} = f(t_n + c_ih_n, Y_i^{(0)}, Z'_0(t_n + c_ih_n - \tau(t_n + c_ih_n, Y_i^0)))$$

Otherwise if $i \leq s_1 + 1$then:

$$K_i^{(0)} = f(t_n + c_ih_n, Y_i^{(0)}, Z'_{i-2}(t_n + c_ih_n - \tau(t_n + c_ih_n, Y_i^0)))$$

Otherwise, we would have:

$$K_i^{(0)} = f(t_n + c_ih_n, Y_i^{(0)}, Z'_{s1}(t_n + c_ih_n - \tau(t_n + c_ih_n, Y_i^0)))$$

Step (4): print the special insert.

Algorithm (2) iteration procedure:

Step (1): for the values of the phases $i = 2,\ldots,\bar{s} + 1$

DOI: https://doi.org/10.54216/GJMSA.0110101
Received: October 20, 2023 Revised: February 25, 2024 Accepted: June 20, 2024

$$Y_i^{\ell} = y_{n-1}(t_n) + h_n a_{i1} K_1 + h_n \sum_{j=2}^{i-1} a_{ij} K_j^{\ell}$$

Step (2): if $t_n < t_n + c_i h_n - \tau(t_n, Y_i^{(0)}) \leq t_{n+1}$

Then we have:

$$K_i^{\ell} = f(t_n + c_i h_n, Y_i^{\ell}, {Z'}_n^{(\ell-1)}(t_n + c_i h_n - \tau(t_n + c_i h_n, Y_i^{(\ell)})))$$

Otherwise, then it will be:

$$K_i^{\ell} = f(t_n + c_i h_n, Y_i^{\ell}, Z'(t_n + c_i h_n - \tau(t_n + c_i h_n, Y_i^{(\ell)})))$$

Step (3): Calculate:

$${Z'}_n^{(\ell)}(t_n + \theta h_n) = y_{n-1}(t_n) + h_n \bar{b}_1(\theta) K_1 + h_n \sum_{i=2}^{\bar{s}+1} \bar{b}_i(\theta) K_i^{\ell}$$

4. Analysis of the convergence of the hybrid method between the Special Interpolation and the iteration procedure of RK

In this item, we will analyze the sufficient conditions that make the proposed hybrid method convergent. In order to arrive at the convergence result, we will assume that the solution is sufficiently smooth except at a specified number of points, and that for time-delayed differential equations of the neutral type and with a vanishing delay (VNDEs) the numerical method is (consistent) by fulfilling the following Condition [4]:

$$z_n(t_n + \theta h_n) = z(t_n) + h_n \sum_{i=1}^{s+1} b_i(\theta) f\left(t_n, z(t_n), z'\left(t_n - \tau\big(t_n, y(t_n)\big)\right)\right) + O(h_n^2)$$

Taken (1) [4]: if the sequence $\{\beta_n(t)\}$achieves the following inequality:

$\beta_n \leq [1 + Lh_n]\beta_{n-1} + (d_n h_n) n = 0,1, \ldots$, since the sequences $\{\beta_n\}$, $\{d_n\}$, $\{h_n\}$ represent non-negative sequences and L is a non-negative constant.

Then: $\beta_n \leq [\, \beta_{-1} + \sum_{i=0}^{n} d_i h_i] \exp\left(\sum_{j=0}^{n} L h_j\right)$

Taken (2): if the functions y' and τ fulfill the following Lipschitz conditions:

1- $\|y'(t_2) - y'(t_1)\| \leq L_{y'} \|t_2 - t_1\|$
2- $\|\tau(t, y_2) - \tau(t, y_1)\| \leq L_T \|y_2 - y_1\|$

Where $L_{y'}$ represents the Lipschitz constant for the derivative of the function y' and L_τ represents the Lipschitz constant for the derivative of the function τ. Then:

$$\|z'(t - \tau(t, z(t))) - y'(t - \tau(t, y(t)))\| \leq \|z'(t - \tau(t, z(t))) - y'(t - \tau(t, z(t)))\| + L_{y'} L_\tau \|z(t) - y(t)\|$$

Proof: applying the trigonometric inequality and using Lipschitz's conditions (1) and (2) above, we have:

$$\begin{aligned} &\|z'(t - \tau(t, z(t))) - y'(t - \tau(t, y(t)))\| + \|y'(t - \tau(t, z(t))) - y'(t - \tau(t, y(t)))\| \\ &\leq \|z'(t - \tau(t, z(t))) - y'(t - \tau(t, z(t)))\| + L_{y'} \|\tau(t, z(t)) - \tau(t, y(t))\| \\ &\leq \|z'(t - \tau(t, z(t))) - y'(t - \tau(t, z(t)))\| + L_{y'} L_\tau \|z(t) - y(t)\| \end{aligned}$$

Now, to analyze the convergence of the hybrid method, we will assume that the functions f, y', and t represent functions that fulfill the Lipschitz conditions and using the consistency condition, we arrive at the convergence of the proposed hybrid method as follows:

From the following formula:

DOI: https://doi.org/10.54216/GJMSA.0110101
Received: October 20, 2023 Revised: February 25, 2024 Accepted: June 20, 2024

$$\begin{cases} u_n(t_n+\theta h_n) = u_{n-1}(t_n) + h_n \sum_{i=1}^{s+1} b_i(\theta) k_i \\ Y_1 = u_{n-1}(t_n) \\ K_1 = f(t_n, Y_1, z'_0(t_n - \tau(t_n, Y_1))) \end{cases}$$

For the value $i = 2, \dots, \bar{s}+1$ we have:

$$\begin{cases} Y_i = u_{n-1}(t_n) + h_n \sum_{j=1}^{i-1} a_{ij} k_j \\ K_i = f(t_n + c_i h_n, Y_i, z'_0(t_n + c_i h_n - \tau(t_n + c_i h_n, Y_i))) \end{cases}$$

And if $t_n + c_i h_n - \tau(t_n + c_i h_n, Y_i) \le t_n + c_i h_n$ then we have:

$$K_i = f(t_n + c_i h_n, Y_i(t_n), z'_0(t_n + c_i h_n - \tau(t_n + c_i h_n, Y_i)))$$

Otherwise we have, if: $i \le s_1 + 1$

$$\therefore\ K_i = f(t_n + c_i h_n, Y_i, z'_{i-2}(t_n + c_i h_n - \tau(t_n + c_i h_n, Y_i)))$$

Otherwise $K_i = f\left(t_n + c_i h_n, Y_i, z'_{s_2}\left(t_n + c_i h_n - \tau(t_n + c_i h_n, Y_i)\right)\right)$, s_1 is the largest number of the special inclusion derivative for each step. Now suppose that:

$$\alpha_n(t) = \|z(t) - y(t)\|, \qquad \beta_n(t) = \max_{t_0 \le \theta \le t_{n+1}} [\alpha(s)]$$

$$z_n(t_n + \theta h_n) - u_n(t_n + \theta h_n)$$

$$= (z(t_n) - u_{n-1}(t_n)) + h_n \sum_{i=1}^{s+1} b_i(\theta)[f(t_n, z(t_n), z'(t_n)$$

$$-\tau(t_n, y(t_n)))) - f(t_n, u_{n-1}(t_n), z'_0(t_n - \tau(t_n, u_{n-1}(t_n))))] + O(h_n^2) \quad ..(1)$$

And for values of $i = 2, \dots, \bar{s}+1$we have:

$$z_n(t_n + \theta h_n) - u_n(t_n + \theta h_n)$$

$$= (z(t_n) - u_{n-1}(t_n)) + h_n \sum_{i=1}^{s+1} b_i(\theta)[f(t_n, z(t_n), z'(t_n)$$

$$-\tau(t_n, y(t_n)))) - f(t_n + c_i h_n, Y_i(t_n), z'_0(t_n + c_i h_n -$$

$$\tau(t_n + c_i h_n, Y_i(t_n))))] + O(h_n^2) \quad ..(2)$$

So: $Y_i(t_n) = u_{n-1}(t_n) + h_n \sum_{j=1}^{i=1} a_{ij} k_j$

If $t_n + c_i h_n - \tau(t_n + c_i h_n) \le t_n + c_i h_n$ then we have the case of Equation (2) itself, while if $i \le s_1 + 1$ it replaces z'_0 in Equation (2) value z'_{i-2} otherwise, if $i > s_1 + 1$ it replaces z'_0 in Equation (2) value z'_{s_1}The s_1 represent the largest number the fall of each step. We will demonstrate the convergence in the case of z'_0in the same way can be the proof in the case z'_{i-2} or z'_{s_1}

$$\|z_n(t_n + \theta h_n) - u_n(t_n + \theta h_n)\|$$

$$\le \|z(t_n) - u_{n-1}(t_n)\|$$

$$+h_n \sum_{i=1}^{s+1} b_i(\theta)[L_1 \|z(t_n) - u_{n-1}(t_n)\|$$

$$+L_3 \left\| z'\left(t_n - \tau(t_n, y(t_n))\right) - z'_0\left(t_n - \tau\left(t_n, u_{n-1}(t_n)\right)\right) \right\|] + O(h_n^2)$$

DOI: https://doi.org/10.54216/GJMSA.0110101
Received: October 20, 2023 Revised: February 25, 2024 Accepted: June 20, 2024

$\leq \beta_{n-1}(t) + h_n \sum_{i=1}^{s+1} b_i(\theta)[L_2\beta_{n-1}(t) + L_3 \left\| z'(t_n - \tau(t_n, y(t_n))) - z'_0\left(t_n - \tau\left(t_n, u_{n-1}(t_n)\right)\right)\right\| + O(h_n^2)$ …(3)

$$\|z_n(t_n + \theta h_n) - u_n(t_n + \theta h_n)\|$$
$$\leq \|z(t_n) - u_{n-1}(t_n)\|$$
$$+h_n \sum_{i=1}^{s+1} b_i(\theta)[L_1\|z(t_n) - Y_i(t_n)\|$$
$$+L_3 \left\| z'\left(t_n - \tau\left(t_n, y(t_n)\right)\right) - z'_0\left(t_n + c_i h_n - \tau\left(t_n + c_i h_n, Y_i(t_n)\right)\right)\right\|] + O(h_n^2)$$
$$\leq \beta_{n-1}(t) + h_n \sum_{i=1}^{s+1} b_i(\theta)[L_1\beta_{n-1}(t) + L_3 \left\| z'(t_n - \tau(t_n, y(t_n))) - z'_0\left(t_n + c_i h_n - \tau\left(t_n + c_i h_n, Y_i(t_n)\right)\right)\right\| + O(h_n^2)$$

Since: $Y_i(t_n) = u_{n-1}(t_n) + h_n \sum_{j=1}^{i=1} a_{ij} K_j$, now suppose:

$\gamma_{(n)}(t) = \max_{t_0 \leq t \leq t_{n+1}} \|z'(t) - y'(t)\|$ since $0 \leq \sum_{i=1}^{s+1} b_i(\theta) \leq 1$ and the right-hand side is greater than the $\beta_{n-1}(t)$ using the Taken (2) we get:

$$h_n[L_1\beta_{n-1}(t) + L_3 \left\| z'\left(t_n - \tau\left(t_n, y(t_n)\right)\right) - z'_0\left(t_n + \tau\left(t_n, u_{n-1}(t_n)\right)\right)\right\| + L_3 L_{y'} L_\tau \beta_{n-1}(t)] + c_1 h_n^2$$

$\therefore \beta_n(t) \leq [1 + h_n(L_1 + L_3 L_{y'} L_\tau)]\beta_{n-1}(t) + h_n L_3 \gamma_{n-1}(t) + c_1 h_n^2$ …(5)

$$\beta_n(t) \leq \beta_{n-1}(t) + h_n[L_1\beta_{n-1}(t)$$
$$+L_3 \left\| z'\left(t_n - \tau\left(t_n, y(t_n)\right)\right) - z'_0\left(t_n + c_i h_n - \tau\left(t_n + c_i h_n, Y_i(t_n)\right)\right) + L_3 L_{y'} L_\tau \beta_{n-1}(t)]\right\| + c_1 h_n^2$$

$\therefore \beta_n(t) \leq [1 + h_n(L_1 + L_3 L_{y'} L_\tau)]\beta_{n-1}(t) + h_n L_3 \gamma_{n-1}(t) + c_1 h_n^2$ …(6)

Since c_1 represents a non-negative number and $Y_i(t_n) = u_{n-1}(t_n) + h_n \sum_{j=1}^{i-1} a_{ij} k_j$ also using taken (2) we have:

$$\|z'_n(t_n + \theta h_n) - u'_n(t_n + \theta h_n)\|$$
$$\leq L_1\|z(t_n) - u_{n-1}(t_n)\|$$
$$+L_3 \left\| z'\left(t_n - \tau\left(t_n, y(t_n)\right)\right) - z'_0\left(t_n - \tau\left(t_n, u_{n-1}(t_n)\right)\right)\right\| + O(h_n)$$
$$\leq L_1\|z(t_n) - u_{n-1}(t_n)\|$$
$$+L_3 \left\| z'\left(t_n - \tau\left(t_n, y(t_n)\right)\right) - z'_0\left(t_n - \tau\left(t_n, u_{n-1}(t_n)\right)\right)\right\| + L_3 L_{y'} L_\tau$$
$$+c_2 h_n \leq (L_1 + L_3 L_{y'} L_\tau)\|z(t_n) - u_{n-1}(t_n)\| + L_3 \gamma_n(t) + c_2 h_n$$

$\leq (L_1 + L_3 L_{y'} L_\tau)\beta_n(t) + L_3 \gamma_n(t) + c_2 h_n$ …(7)

c_2 is a non-negative number. As well as:

$$\|z'_n(t_n + \theta h_n) - u'_n(t_n + \theta h_n)\|$$
$$\leq L_1\|z(t_n) - Y_i(t_n)\|$$
$$+L_3 \left\| z'\left(t_n - \tau\left(t_n, y(t_n)\right)\right) - z'_0\left(t_n + c_i h_n - \tau\left(t_n + c_i h_n, Y_i(t_n)\right)\right)\right\| + O(h_n)$$
$$\leq (L_1 + L_3 L_{y'} L_\tau)\|z(t_n) - Y_i(t_n)\| + L_3 \gamma_n(t) + c_2 h_n$$

$\leq (L_1 + L_3 L_{y'} L_\tau)\beta_n(t) + L_3 \gamma_n(t) + c_2 h_n$ …(8)

DOI: https://doi.org/10.54216/GJMSA.0110101
Received: October 20, 2023 Revised: February 25, 2024 Accepted: June 20, 2024

From equations (7) and (8), we conclude that in both cases, whether u_{n-1} or Y_i, we have:

$\|z'_n(t_n + \theta h_n) - u'_n(t_n + \theta h_n)\| \leq (L_1 + L_3 L_{y'} L_\tau)\beta_n(t) + L_3\gamma_n(t) + c_2 h_n$ …(9)

Now we assume that:

$$\gamma_n(t) = \|z'_i(t_i + \theta h_i) - u'_i(t_i + \theta h_i)\|$$
$$\leq (L_1 + L_3 L_{y'} L_\tau)\beta_i(t) + L_3\gamma_i(t) + c_2 h_i$$
$$\therefore \gamma_n(t) \leq (L_1 + L_3 L_{y'} L_\tau)\beta_n(t) + L_3\gamma_n(t) + c_2 H$$

Since $H = \max_i h_i$

$$\therefore \gamma_n(t) - L_3\gamma_n(t) \leq (L_1 + L_3 L_{y'} L_\tau)\beta_n(t) + c_2 H$$
$$(1 - L_3)\gamma_n(t) \leq (L_1 + L_3 L_{y'} L_\tau)\beta_n(t) + c_2 H$$

$\therefore \gamma_n(t) \leq \frac{1}{1-L_3}[(L_1 + L_3 L_{y'} L_\tau)\beta_n(t)] + \frac{1}{1-L_3}c_2 H$ ..(10)

Suppose that $L = L_1 + L_3 L_{y'} L_\tau$ and substituting equation (10) with Equation (5) or (6) we have:

$$\beta_n(t) \leq (1 + h_n L)\beta_{n-1}(t) + h_n L_3 \frac{1}{1 - L_3}\beta_{n-1}(t) + h_n L_3 \frac{1}{1 - L_3} c_2 H$$
$$+c_1 h_n^2 \leq (1 + \frac{1}{1 - L_3} h_n)\beta_{n-1}(t) + (c_1 h_n + \frac{c_2 L_3 H}{1 - L_3})h_n$$
$$\leq (1 + \frac{1}{1 - L_3} h_n)\beta_{n-1}(t) + (c_1 H + \frac{c_2 L_3 H}{1 - L_3})h_n \quad since \;\; H = \max_n h_n$$
$$\leq (1 + \frac{1}{1 - L_3} h_n)\beta_{n-1}(t) + (c_1 + \frac{c_2 L_3 H}{1 - L_3})h_n H$$

Now from taken (1), [4] we have:

$$\beta_n(t) \leq [\beta_{-1} + \sum_{i=0}^{n}(c_1 + \frac{c_2 L_3}{1 - L_3})h_i H]exp(\sum_{j=0}^{n}\frac{L}{1 - L_3}h_j) \leq TH(c_1 + \frac{c_2 L_3}{1 - L_3})exp(\frac{LT}{1 - L_3})$$

Since $T = \sum_{i=0}^{n} h_i$, therefore, when $H \to 0$, $\beta_n(t) \to 0$, We conclude that the solution is convergent. From equation (10) we conclude that the derivative of the solution is convergent as well. That is:

$$\therefore \gamma_n(t) \leq \frac{H}{1 - L_3}[LT(c_1 + \frac{c_2 L_3}{1 - L_3})exp(\frac{L\,T}{1 - L_3}) + c_2]$$

And when H→0, $\gamma_n(t) \to 0$

5. Practical application

5.1. Set of problems: the set of problems described below is taken from the source [4] and is considered one of the test problems of time-delayed differential equations. These solutions have been chosen either for their smooth form or for having the discontinuity of the derivative at some points in the range of integration. It has been observed that all the selected problems possess a delay at both solutions and derivatives.

<u>The first issue</u>

$$y''(t) = Ay'(t - B) \qquad t \geq 0$$
$$Y(t) = C \qquad t \leq 0$$

DOI: https://doi.org/10.54216/GJMSA.0110101
Received: October 20, 2023 Revised: February 25, 2024 Accepted: June 20, 2024

The exact solution: $\boldsymbol{y(t) = C\sum_{n=0}^{[\frac{t}{B}]+1} A^n \frac{(t-(n-1)B)^n}{n!}}$ $\quad\quad \boldsymbol{t \geq 0}$

Other information: for Values A=B=C=1 the solution at Value t=10 is $\boldsymbol{y(10)} = \frac{\mathbf{14640251}}{\mathbf{44800}}$

For the values A= -1 and B=C=1, the solution hind t=10 is $\boldsymbol{y(10)} = \frac{\mathbf{10493}}{\mathbf{518400}}$

The second issue

$$y_1'(t) = 3.05 - 0.1exp\left(-0.02y_3(t)\right)$$

$$-(1-\tau'(t)).(3.05 - 0.1exp(-0.02y_3(t-\tau(t)) \quad\quad t \geq 0$$

$$y_2'(t) = (1-\tau'(t)).(3.05 - 0.1exp(-0.02y_3(t-\tau(t)) - y_2(t)) \quad t \geq 0$$

$$y_3'(t) = 4.2\ exp(-0.05y_2(t)) - y_3(t) \quad t \geq 0$$

$$Y(t) = [1,0.1,1]^T$$

Since $\tau(t) = 0.2 + 1.2\exp\left(-0.001\ y_3(t)\right)$

Other information: the exact solution is $y_1(20) = 0.997721687800035$

$y_2(20) = 2.95698924553131$and $y_2(20) = 2.95698924553131$

The third issue

$$y'(t) = cos(t)\ (1 + y(ty(t)^2)) + y(t)\ y'(ty(t)^2) - sin\ (t + t\ sin\ (t)^2) \quad\quad 0 \leq t \leq \frac{\pi}{2}$$

$Y(0) = 0$

$Y'(0) = 1$

The exact solution is $0 \leq t \leq \frac{\pi}{2}$ $\quad \boldsymbol{y(t) = sin(t)}$

5.2. Developer ARCHI, ARCHI full description of the two programs

ARCHI [2] (1)

The formula of this program (code): is built based on the Rang –Kota-explicit method of the fifth rank of Domand and Prince with an insertion of the Quadric type of the fifth rank.

How to deal with discontinuity: there are two processes either with a tracking path or without tracking, and to use the path process, the one used must concern the dependence of the discontinuity of the derivative among the compounds of time-delayed differential equations, as well as the location when the derivatives are discontinuous in the initial period and the rank of discontinuity of derivatives [2]. Thus, the code determines the non-propagation of discontinuities by finding the zeros of the related nonlinear equations shown based on the method proposed by Wille, Baker. The matter or process without a path is assigned by the error control mechanics to the formula referred to in this program Code.

For Vanishing delay there are two operations either completion or inclusion.

(2) ARCHI developer:

The formula of the new proposed program (Code): this proposed formula is built on the basis of using the hybrid method between special inclusion and iterative methods of RK to solve problems (VNDEs) and built on the basis of using the Rang-Kota-explicit method of Prince, Dormond used in the old ARCHI program but by forming preliminary approximations or special inclusion based on all available information from the previous step and the first stages of the current calculated step. Moreover, this inclusion is built based on two times, and then use this new inclusion with the iterative methods of the RK method.

Vanishing delay: the occurrence of delay in the current calculated step due to the delay being smaller than the step size h or may sometimes reach vanishing i.e. τ→0.

DOI: https://doi.org/10.54216/GJMSA.0110101
Received: October 20, 2023 Revised: February 25, 2024 Accepted: June 20, 2024

5.3. Numerical results

In this section, we show the numerical results of the three problems that we obtained, as they were summarized through tables showing those results and the resulting error. The numerical results indicate that the proposed new program is more powerful and efficient with various types of time-delayed differential equations of the neutral type with vanishing delay (VNDEs) compared to the old ARCHI program.

The first issue:

Table 1: The amount of error and the solution at (Xend = 10) using the two programs ARCHI and Improved ARCHI to solve the first issue

the amount of relative error (Improved ARCHI)	the amount of relative error (ARCHI)	the value of y at Xend (Improved ARCHI)	the value of y at Xend (ARCHI)	ToL
-1.338640309711536E-011	5.671431564380214E-05	326.791316959911	326.809850710186	10^{-10}
-1.338694710639743E-10	5.671419512220943E-05	326.791316920538	326.809850670800	10^{-9}
-1.899848189879094E-08	5.669533160146401E-05	326.791310755747	326.809844506365	10^{-7}
-5.261744620488340E-05	-2.156847192247291E-05	326.774122039745	326.784268574941	$\mathbf{10^{-5}}$
-1.072532158120887E-02	-1.089374933994247E-02	323.286374999897	323.231334270807	$\mathbf{10^{-3}}$

The second issue:

Table 2: Tthe amount of error and the solution y_1 at (Xend = 20) using the two programs

the amount of relative error (Improved ARCHI)	the amount of relative error (ARCHI)	the value of $y_1$at Xend (Improved ARCHI)	the value of y_1 at Xend (ARCHI)	ToL
1.896192136641162E-08	2.957201061803971E-08	0.997721706718755	0.997721717304671	10^{-10}
1.645212091005988E-07	3.414536913215471E-07	0.997721851946413	0.997722028475788	10^{-9}
1.956920048451494E-06	-9.391057986696261E-06	0.997723640261609	0.997712318137810	10^{-7}
6.246361963979652E-05	-1.003882604313500E-03	0.997784009108048	0.996720092353706	$\mathbf{10^{-5}}$
-0.120949259975030	-7.568106060456182E-03	0.877047987999583	0.990170824257884	$\mathbf{10^{-3}}$

Table 3: The amount of error and solution y_2 at (Xend = 20) using the two programs ARCHI and Improved ARCHI to solve the second issue

the amount of relative error (Improved ARCHI)	the amount of relative error (ARCHI)	the value of $y_2$at Xend (Improved ARCHI)	the value of y_2 at Xend (ARCHI)	ToL
-9.426237568277429E-012	-3.047628815977532E-011	2.95698924550344	2.95698924544119	10^{-10}
-1.020933337869678E-010	-7.030853677036930E-011	2.95698924522942	2.95698924532341	10^{-9}
-1.637299962276018E-009	-1.675896532660204E-009	2.95698924068983	2.95698924057570	10^{-7}
3.222635158728338E-08	-2.815976041903312E-06	2.95698934082429	2.95698091872044	$\mathbf{10^{-5}}$
8.988221009764708E-05	-1.416647810170701E-06	2.95725502625993	2.95698505651897	$\mathbf{10^{-3}}$

Table 4: The amount of error and solution y_3 at (Xend = 20) using the two programs ARCHI and Improved ARCHI to solve the second issue

the amount of relative error (Improved ARCHI)	the amount of relative error (ARCHI)	the value of y_3 at Xend (Improved ARCHI)	the value of y_3 at Xend (ARCHI)	ToL
6.119771356338788E-012	1.778310831923591E-011	3.62275601806450	3.62275601810675	10^{-10}
5.514966261443988E-011	4.612199511200288E-011	3.62275601824212	3.62275601820942	10^{-9}
1.061315035855159E-09	1.353416045901668E-09	3.62275602188722	3.62275602294543	10^{-7}
-7.647850297498593E-08	-4.236490460329811E-06	3.62275574097937	3.62274067027102	$\mathbf{10^{-5}}$
4.628785248664235E-04	-4.411177573071523E-05	3.62421554864283	3.62259621184134	$\mathbf{10^{-3}}$

DOI: https://doi.org/10.54216/GJMSA.0110101
Received: October 20, 2023 Revised: February 25, 2024 Accepted: June 20, 2024

The third issue:

Table 5: The amount of error and the solution at (Xend = 1) using the two programs ARCHI and Improved ARCHI to solve the third issue

the amount of relative error (Improved ARCHI)	the amount of relative error (ARCHI)	the value of y_2 at Xend (Improved ARCHI)	the value of y_2 at Xend (ARCHI)	ToL
7.623983444560167E-09	1.000020188546280E-04	0.841470991223257	0.841555133605185	10^{-10}
1.813368883940569E-07	9.992740381026266E-05	0.841471137397627	0.841555070818790	10^{-9}
9.676552606416886E-06	9.312532403416007E-05	0.841479127346147	0.841549347066022	10^{-7}
1.287041587800089E-03	-6.543522469970586E-05	0.842553992539536	0.841415922964927	$\mathbf{10^{-5}}$
-4.347237677155835E-04	-3.347506533729616E-04	0.841105177370957	0.841189301845938	$\mathbf{10^{-3}}$

Tables 5 shows that the proposed new program has greater accuracy and less error than the old ARCHI program, so the hybrid method is characterized by strength and efficiency with various types of time-delayed differential equations of the VNDEs type.

6. Conclusion

The main contribution of this research is the construction of a hybrid method between the special inclusion and the iterative method of the RK method used to solve VNDEs problems and the convergence analysis of the proposed method. We concluded that the hybrid method between the special insertion and the iterative method of the RK method can be adapted to solve VNDEs problems and that the efficiency of this method was demonstrated by applying the proposed new method described by algorithms (1) and (2) to the ARCHI program. The developed ARCHI program has proven its high efficiency and accuracy of results compared to the original ARCHI program. In this research, the convergence of the hybrid method was also analyzed, and it was concluded that the convergence measure of this proposed hybrid method is:

$$\beta_n(t) \leq TH\left(c_1 + \frac{c_2 L_3}{1 - L_3}\right) exp\left(\frac{LT}{1 - L_3}\right)$$

And that when H→0, $\beta_n(t) \to 0$, the solution is convergent, and the derivative of the solution is convergent as well in the case of the vanishing delay of (VNDEs) problems.

References

[1] Enright W. H., Hu M. (1994), interpolating Runge –Kutta methods for vanishing delay differential equations. Numerical Algorithm, Vol.16.

[2] Faires J. D., Burden R. L. (2003), Numerical Methods-Third Edition. Inc. Thomson Learning. Australia. Canada. Mexico. Singapore. Spain. United Kingdom. United States.

[3] Fu H., Li S. , Chen G. (2001) , D -Convergence of Runge -Kutta Method for Stiff Delay Differential Equations. Journal of Computational Mathematics, Vol.19, No.3, pp.: 259-268.

[4] Hayashi H. (1996), Numerical Solution of Retarded and Neutral Delay Differential Equations using RK methods. University of Toronto, Department of Computer Sciences, Ph.D. Thesis

[5] Paul C. A. H. (1997), A user guide to ARCHI. University of Manchester/ UMIST Manchester Center for Computational Mathematics, England, Vol.283, ISSN: 1360-1725.

[6] Paul C. A. H. (1991), Developing a Delay Differential Equation Solver. University of Manchester /UMIST Manchester Center for Computational Mathematics, England, Vol.204, ISSN: 1360-1725.

[7] Shampine L. F., Thompson S. (2000), Solving Delay Differential Equations with DDE23. Southern Methodist University, March 23.

[8] Shampine L. F., Thompson S. (2001), Solving DDEs in MATLAB. Applied Numer. Math. , Vol.37, No.4 pp.: 441-458.

[9] Neves K.W. (1981), Control of interpolatory error in retarded differential equations. ACM Trans. Math.Soft. , 7:421-444.

[10] ackiewict Z. (1981), One step methods for the numerical solution of volterra functional differential equations of neutral type. Applicable Anal., 12:1-11.

DOI: https://doi.org/10.54216/GJMSA.0110101
Received: October 20, 2023 Revised: February 25, 2024 Accepted: June 20, 2024