

ORIGINAL ARTICLE OPEN ACCESS

# Using Multidelay Discrete Delay Differential Equations to Accurately Simulate Models With Distributed Delays

Tyler Cassidy 

School of Mathematics, University of Leeds, Leeds, UK

Correspondence: Tyler Cassidy ([t.cassidy1@leeds.ac.uk](mailto:t.cassidy1@leeds.ac.uk))

Received: 22 January 2025 | Revised: 25 March 2025 | Accepted: 19 May 2025

Funding: The author received no specific funding for this work.

Keywords: delay differential equations | distributed delay differential equations | numerical analysis

## ABSTRACT

Delayed processes are ubiquitous throughout biology. These delays may arise through maturation processes or as the result of complex multistep networks, and mathematical models with distributed delays are increasingly used to capture the heterogeneity present in these delayed processes. Typically, these distributed delay differential equations are simulated by discretizing the distributed delay and using existing tools for the resulting multidelay delay differential equations or by using an equivalent representation under additional assumptions on the delayed process. Here, we use the existing framework of functional continuous Runge–Kutta methods to confirm the convergence of this common approach. Our analysis formalizes the intuition that the least accurate numerical method dominates the error. We give a number of examples to illustrate the predicted convergence, derive a new class of equivalences between distributed delay and discrete delay differential equations, and give conditions for the existence of breaking points in the distributed delay differential equation. Finally, our work shows how recently reported multidelay complexity collapse arises naturally from the convergence of equations with multiple discrete delays to equations with distributed delays, offering insight into the dynamics of the Mackey–Glass equation.

## 1 | Introduction

Delayed processes, such as maturation or circadian processes [1–3] or transcriptional regulation [4–6], are commonly found throughout biology and increasingly modeled using delay differential equations (DDEs) [7]. While these DDEs typically use discrete delays to capture the influence of the past state on the current dynamics, it is increasingly understood that heterogeneity in the duration of these delayed processes can be an important factor in understanding biological dynamics [8, 9]. Consequently, there has been increased interest in distributed DDEs that, rather than considering the state at discrete times in the past, consider a continuum of past states typically weighted against a probability distribution through a convolution integral. Many recent examples consider distributed DDEs where

the delay is compactly supported; both Sargood et al. [5] and Hong et al. [4] used compactly supported distributed DDEs to understand the dynamics of a gene regulatory system. These compactly supported distributed DDEs are generically given by

$$\left. \begin{aligned} \frac{d}{dt}x(t) &= F\left(x(t), \int_{\tau_{\min}}^{\tau_{\max}} x(t-s)k(s)ds\right) \\ x(s) &= \varphi(s) \quad \text{for } s \in [-\tau_{\max}, 0] \end{aligned} \right\} \quad (1.1)$$

where  $F$  is Lipschitz,  $k(s)$  is a continuous function integrates to unity (possibly after a scaling) on the interval  $[\tau_{\min}, \tau_{\max}]$ , and  $0 < \tau_{\min} < \tau_{\max} < T < \infty$ .

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2025 The Author(s). *Studies in Applied Mathematics* published by Wiley Periodicals LLC.

To simulate the distributed DDE, modelers commonly use a quadrature method to discretize the convolution integral in Equation (1.1) and thus obtain a multidelay discrete DDE. Then, modelers use existing methods for discrete DDEs to simulate the resulting system [5, 10]. However, the quadrature method and step size in the discretization of the convolution integral are often chosen intuitively. For example, both [11, 12] discretized a distributed delay term in the bifurcation analysis of a state-dependent DDE model of transcriptional delays where the choice of discretization parameters was described as delicate. Ultimately, an analysis of how the error arising from the quadrature method interacts with the existing numerical method for the DDE in these models would need to be completed. To avoid this problem, some authors [13–15] have used approaches similar to the linear chain technique to show that, for specific choices of the kernel  $k$ , the distributed DDE is equivalent to a system of discrete DDEs. We derive a similar equivalence for kernels representing a mixture of exponential-type distributions. However, these approaches are unsatisfying as they rely on artificial assumptions on the kernel  $k$ .

Therefore, we develop a numerical method for the simulation of Equation (1.1) based on existing functional continuous Runge–Kutta (FCRK) methods. These FCRK methods were proposed in the 1970s [16, 17], have been implemented for distributed DDEs with possibly time-dependent, but finite, delay [18, 19], and recently extended to gamma distributed DDEs [20]. The convergence theory for FCRKs for equations with fixed delays was derived in the 2000s [17, 21] along with a specific distributed DDE formulation as an example of the convergence framework [21]. Here, we use these ideas to show the convergence of the common discretization method used for models with distributed DDEs with compact support. We derive an explicit relationship between the error introduced by approximating the convolution integral via a quadrature method with the numerical method for the time integration of the DDE. This relationship quantifies the intuitive expectation that the accuracy of the simulation is determined by the least accurate of the quadrature and Runge–Kutta methods. As a result, we show how accurate the quadrature method must be to preserve the overall accuracy of the numerical simulations. While this result is intuitive, establishing the precise relationship between the error of the quadrature method and the error of the Runge–Kutta method will be useful for modelers who use distributed DDEs. Specifically, the conditions on the quadrature method will allow distributed DDEs to be accurately simulated via existing numerical methods for discrete DDEs without model-specific analysis. Consequently, this convergence framework will allow modelers to a priori choose the appropriate discretization method for a distributed DDE model without requiring problem-specific analysis.

We illustrate these theoretical results through a number of examples of distributed DDEs with compact support. As is often the case, our convergence proof depends on detecting breaking points of the distributed DDE (1.1). At these breaking points, the solution  $x$  does not have continuous derivatives of all orders, which may influence the error estimates of numerical methods. In practice, it is well known that breaking points in discrete DDEs result in a decrease of convergence order of numerical methods [22]. We therefore give an explicit condition on the kernel  $k$  that ensures that the breaking point at  $t = 0$

does not propagate forward in time. For example, the uniform distribution does not satisfy this condition and is an explicit case of a distributed DDE with breaking points. We observe the corresponding decrease in convergence order due to these breaking points in our numerical examples.

Our convergence result explicitly demonstrates that distributed DDEs with compact support are numerically equivalent to multidelay discrete DDEs. Importantly, this numerical equivalence also applies in the opposite direction. Tavakoli and Longtin [10] observed that, for a number of prototypical DDEs, including the Lang–Kobayashi and Mackey–Glass equations, increasing the number of delays increases dynamical complexity until dynamics suddenly simplify. The authors noted that this complexity collapse is paradoxical, as increasing the number of delays in a discrete DDE typically *increases* the complexity of the resulting dynamics, rather than leading to the observed complexity collapse. Here, we resolve this apparent paradox by using the ideas underlying the convergence of the numerical method to demonstrate that this complexity collapse is due to convergence of the multidelay discrete DDE converges to a distributed DDE with compact support.

The remainder of the article is structured as follows. We first describe FCRK methods for distributed DDEs and establish a consistent FCRK for distributed DDEs with compact support. We illustrate this convergence result through a number of numerical examples that use the Matlab built-in DDE solver, `ddesd` [23, 24]. We then show that the multidelay complexity collapse for the Mackey–Glass equation [10] can be understood as an example of a distributed DDE having simpler behavior than a multidelay DDE.

## 2 | FCRK Methods

A class of numerical methods, based upon the familiar Runge–Kutta methods for ordinary differential equations, has been developed for DDEs [17]. These FCRKs define a continuous interpolant within each Runge–Kutta step. A number of these FCRK methods have been developed [19, 20, 25]. We give only the notation necessary for the convergence result with detailed descriptions of these methods available elsewhere [17, 26]. We define an  $s$ -stage FCRK method following Definition 6.1 of [17].

**Definition 2.1** ( $s$ -stage FCRK method). A  $s$ -stage FCRK method is a triple  $(A(\theta), b(\theta), c)$  such that  $A$  and  $b$  are polynomial functions into  $\mathbb{R}^{s \times s}$  and  $\mathbb{R}^s$ , respectively, with  $A(0) = 0$  and  $b(0) = 0$ , and  $c \in \mathbb{R}^s$  with  $c_i \geq 0$ .

This  $s$ -stage FCRK method  $(A(\theta), b(\theta), c)$  can be represented by its Butcher tableau

$$\begin{array}{c|c} c_i & A_{i,j}(\theta) \\ \hline & b_j(\theta) \end{array},$$

where  $i, j = 1, 2, 3, \dots, s$  and  $A_{i,j}$  and  $b_j$  are the components of  $A$  and  $b$ . For a given step size  $h$ , the  $s$ -stage FCRK method creates a continuous approximation  $\eta(t)$  of the solution of the IVP (1.1)

$x(t)$  through

$$\eta(t) = \begin{cases} \eta^n(h\theta) & \text{for } t \in (t_n, t_{n+1}), \text{ and } \theta = \frac{t-t_n}{h} \\ \varphi & \text{for } t \leq t_0. \end{cases} \quad (2.1)$$

The stage interpolant  $\eta^n$  is a continuous approximation of the solution  $x(t_n + h\theta)$  over the  $n$ th stage and is defined by

$$\begin{aligned} \eta^n(h\theta) &= x^n + h \sum_{i=1}^s b_i(\theta) K_{n,i}, \quad \theta \in (0, 1) \quad \eta^0 = \varphi(t_0), \\ \text{and } x^n &= \eta^{n-1}(h), \end{aligned} \quad (2.2)$$

where

$$K_{n,i} = F \left( Y^{n,i}(c_i), \int_{\tau_{min}}^{\tau_{max}} \eta(t_n + c_i h - u) k(u) du \right) \quad (2.3)$$

are the stage variables,  $\eta$  represents the numerical approximation of the solution up to the current stage, and  $Y^{n,i}$  is the continuous approximation of  $x(t)$  in the stage given by

$$Y^{n,i} = x^n + h \sum_{k=1}^{i-1} A_{i,k}(\theta) K_{n,k}, \quad \theta \in [0, c_i].$$

These piecewise interpolants  $\eta^n(t)$  agree with  $x^n$  at the collocation points  $t = \{jh\}_{j=1}^N$  and define the piecewise continuous polynomial function  $\eta$ . For Equation (1.1) with history function  $\varphi$  and step size  $h$  computed up to  $t_m$ , the local error function is given by

$$E(a, t_m, \varphi) = \|\eta(t_m + a) - x(t_m + a)\|, \quad a \in [0, h].$$

The uniform order of a FCRK method is intrinsically related to this local error function [21] as the maximal error incurred over a single time step.

**Definition 2.2** (Uniform order). Let  $d$  be a positive integer and let  $\eta$  be the approximation of the solution  $x$  of an IVP with sufficiently smooth right-hand side obtained using an FCRK method with step size  $h$ . The FCRK method has uniform order  $d$  if

$$\max_{\alpha \in (0,1)} E(\alpha h, t_m, \varphi) = \mathcal{O}(h^{d+1}).$$

Conversely, the discrete order of an FCRK method is the error incurred at the collocation points  $t = jh$ , which corresponds to  $a = h$  in the definition of  $E$ .

**Definition 2.3** (Discrete order). Let  $d$  be a positive integer and let  $\eta$  be the approximation of the solution  $x$  of an IVP with sufficiently smooth right-hand side obtained using an FCRK method with step size  $h$ . The FCRK method has discrete order  $d$  if

$$E(h, t_m, \varphi) = \mathcal{O}(h^{d+1}).$$

The global order of the numerical method is the absolute error incurred throughout the simulation when considering the

solution  $x$  and  $\eta$  as continuous functions on the interval  $t \in [t_0, T]$ .

**Definition 2.4** (Global order). An  $s$ -stage method has global order  $p$  if

$$\max_{t \in [t_0, T]} \|\eta(t) - x(t)\| = \mathcal{O}(h^p).$$

If the  $s$ -stage method has global order  $p$  on  $[t_0, T]$ , then  $\eta$  is a  $p$ th order approximation of  $x$  as [17, 26]

$$\max_{t \in [t_0, T]} \|\eta(t) - x(t)\| < Ch^p.$$

## 2.1 | Quadrature Approximations of Delayed Term

Existing FCRK methods are, in theory, directly applicable to the distributed DDE case under the assumption of being able to accurately calculate the right-hand side of Equation (1.1). In what follows, we assume that we are using a known, existing FCRK method of order  $p$ , such as those considered in [21]. The main difficulty in adapting these FCRK methods to distributed DDEs such as Equation (1.1) is the numerical calculation of the convolution integral

$$I(t) = \int_{\tau_{min}}^{\tau_{max}} x(t-s)k(s)ds. \quad (2.4)$$

At each Runge–Kutta step, we must numerically evaluate this convolution integral where the integrand explicitly depends on  $x(t)$  over the interval  $[\tau_{max}, \tau_{min}]$ . Now, consider an FCRK method where the corresponding interpolant  $\eta$  is an order  $p$  approximation of the solution  $x$  in each stage. As the history function  $\varphi$  is assumed to be evaluated exactly, we calculate

$$\begin{aligned} |(X * k)(t_n) - (\eta * k)(t_n)| &= \left| \int_{t_n - \tau_{min}}^{t_n - \tau_{max}} (x(s) - \eta(s))k(t_n - s)ds \right| \\ &\leq \|x(s) - \eta(s)\|_{L_\infty[t_0, T]} \int_{t_n - \tau_{min}}^{t_n - \tau_{max}} k(t_n - s)ds \\ &\leq Ch^p \int_{t_n - \tau_{min}}^{t_n - \tau_{max}} k(t_n - s)ds = Ch^p. \end{aligned}$$

Now, if we were to calculate  $I(t)$  exactly, then we would evaluate the right-hand side of Equation (1.1) to order  $p$  due to the error introduced via the Runge–Kutta steps.

However, we do not wish to evaluate the convolution integral Equation (2.4) exactly. As the numerical solution  $\eta^n$  is a  $p$ th order approximation of  $x(t)$ , it is not computationally efficient to evaluate the convolution integral  $I(t)$  to precision beyond order  $p$ , as this additional accuracy is washed-out by the interpolant error. Rather, a composite quadrature method should be sufficiently accurate to preserve the global order of the method, but not so accurate as to be computationally inefficient. To illustrate this balance, assume that we evaluate  $I(t)$  to order  $q$  using a composite

quadrature method with step size  $h_{int}$ , so

$$I(t) = \hat{I}(t) + \mathcal{O}(h_{int}^q),$$

where  $\hat{I}(t)$  denotes the quadrature approximation of the convolution integral. Then, we use Taylor's theorem to find

$$\begin{aligned} \hat{K}_{n,1} &= F(x^{n-1}, \hat{I}(t_{n-1})) = F(x^{n-1}, I(t_{n-1}) + \mathcal{O}(h_{int}^q)) \\ &= F(x^{n-1}, I(t_{n-1})) + \partial_{x_2} F(x^{n-1}, I(t_{n-1}))\mathcal{O}(h_{int}^q) + \mathcal{O}(h_{int}^{2q}) \\ &= K_{n,1} + \mathcal{O}(h_{int}^q), \end{aligned} \tag{2.5}$$

where  $\partial_{x_2} F$  is the partial derivative of  $F$  with respect to the second argument. Consequently, the first-stage step  $\hat{Y}_1$  has the same accuracy as the quadrature method and we can approximate each  $\hat{Y}_i$  and  $\hat{K}_i$  with accuracy  $\mathcal{O}(h_{int}^q)$ . Each evaluation of  $F$ , and thus  $I(t)$ , occurs within the calculation of  $K_{n,i}$ , so we therefore gain an extra order of accuracy due to the factor  $h$  in Equation (2.2), where we recall that  $h$  is the step size of the FCRK method. Then, denoting the approximate interpolant obtained via the quadrature approximation by  $\hat{\eta}$ , we find

$$\hat{\eta}^n(h\theta) = x^n + h \sum_{i=1}^s b_i(\theta)\hat{K}_{n,i} = x^n + h \sum_{i=1}^s b_i(\theta)K_{n,i} + \mathcal{O}(h \times h_{int}^q).$$

Therefore, if  $h_{int}^q = \xi h^p$  for some constant  $\xi$ , then  $\mathcal{O}(h \times h_{int}^q) = \mathcal{O}(h^{p+1})$ , which implies that the approximation error in  $\hat{\eta}$  does not influence the accuracy of the method. This analysis suggests setting  $h_{int}^q = \mathcal{O}(h^p)$  to ensure we do not decrease the accuracy of the FCRK nor perform extra computations through the quadrature approximation of  $I(t)$  when using a  $q$ th order, composite quadrature rule.

These quadrature rules typically rely on the integrand being sufficiently smooth, we must include the simulation mesh points  $t_n$  preceding the current step in the quadrature mesh. Therefore, to ensure the global accuracy of the overall FCRK method, we divide the integration domain  $[\tau_{max}, \tau_{min}]$  into subintervals of maximal length  $h_{int} = h$  that include the simulation mesh points and any breaking points. We can then utilize results from [17, 21] to show

**Theorem 2.5** (Global order of the FCRK method). *Assume that the right-hand side of Equation (1.1) is four times continuously differentiable and let  $(A(\theta), b(\theta), c)$  be an explicit FCRK method with global  $p$ th order. Assume that the simulation mesh includes all breaking points of the DDE (1.1) and has a maximal step size  $h_\Delta$ . If the convolution integral  $I(t)$  is calculated using a composite quadrature rule of order  $q$  with maximal subinterval size of  $h_{int} \leq h_\Delta$ , then the resulting FCRK method has global order  $\min(p, q)$ .*

## 2.2 | Convergence of the Resulting FCRK Method

Here, we show that the quadrature approach described in Section 2.1 is sufficient to maintain the convergence of existing FCRK methods described in Section 2 and prove Theorem 2.5. We make extensive use of the results in Section 6 of [21] and Section 7 of [17]. We also recall Definitions 2.2, 2.3, and Definition 4.1 of [21]. Here, we are considering the influence of using quadrature

approximations in existing FCRK methods which corresponds precisely to the setting of Section 6 of [21]. There, the authors considered FCRK methods for the generalized setting of all approximations of the right-hand side of Equation (1.1), of which the quadrature method considered here is an explicit example [21]. In the following analysis, we consider a known FCRK method and focus on the quadrature method. We follow [21] and denote the approximated right-hand side of Equation (1.1) with a tilde

$$\begin{aligned} \tilde{F} \left( x(t), \int_{\tau_{min}}^{\tau_{max}} x(t-s)k(s)ds, \lambda \right) \\ \approx F \left( x(t), \int_{\tau_{min}}^{\tau_{max}} x(t-s)k(s)ds \right), \end{aligned}$$

where  $\lambda \in \Lambda$  is a parameter that controls the precision of the approximation. In our setting,  $\Lambda$  represents the space of composite quadrature rules with a fixed number  $M$  of steps. These quadrature rules are defined by their weights,  $\sigma_i$ , and collocation points,  $\pi_i \in (\tau_{max}, \tau_{min}]$ . The quadrature rule is therefore represented by  $\lambda = (\sigma_1, \dots, \sigma_m, \pi_1, \dots, \pi_m)$  with

$$\int_{\tau_{min}}^{\tau_{max}} x(t-s)k(s)ds = \sum_{i=1}^M \sigma_i x(t-\pi_i)k(\pi_i) + \mathcal{O}(h_{int}^q),$$

where  $h_{int}$  and  $q$  are the step size and order of the composite quadrature method, respectively. Accordingly, the approximation of the right-hand side of Equation (1.1) is given by

$$\tilde{F} \left( x(t), \sum_{i=1}^M \sigma_i x(t-\pi_i)k(\pi_i), \lambda \right) = F \left( x(t), \sum_{i=1}^M \sigma_i x(t-\pi_i)k(\pi_i) \right). \tag{2.6}$$

For a given continuous function  $\varphi$  and quadrature rule  $\lambda$ , we denote the accuracy of the approximation  $\tilde{F}$  by  $\varepsilon(\varphi, \lambda)$ , given by

$$\begin{aligned} \varepsilon(\varphi, \lambda) &= \left| \tilde{F} \left( \varphi(t), \sum_{i=1}^M \sigma_i \varphi(t-\pi_i)k(\pi_i), \lambda \right) \right. \\ &\quad \left. - F \left( \varphi(t), \int_{\tau_{min}}^{\tau_{max}} \varphi(t-s)k(s)ds \right) \right|. \end{aligned} \tag{2.7}$$

The following conditions on  $\tilde{F}$  to ensure the convergence of the FCRK method for Equation (1.1) [21]:

**Condition 1.**  $\tilde{F}(\varphi(t), \sum_{i=1}^M \sigma_i x(t-\pi_i)k(\pi_i), \lambda)$  is uniformly continuous with respect to  $\lambda$  and the derivative with respect to the function  $\varphi$ ,  $\tilde{F}'(\varphi(t), \sum_{i=1}^M \sigma_i x(t-\pi_i)k(\pi_i), \lambda)$  is continuous with respect to  $\varphi$  and uniformly bounded with respect to  $\lambda$ ;

**Condition 2.** There exists a continuous function  $p : C_0([\tau_{max}, \tau_{min}]) \rightarrow \mathbb{R}$  such that

$$\left| \tilde{F} \left( \varphi(t), \sum_{i=1}^M \sigma_i x(t-\pi_i)k(\pi_i), \lambda \right) \right| < p(\varphi)$$

for all  $\varphi \in C_0([\tau_{max}, \tau_{min}])$  and  $\lambda \in \Lambda$ ;

**Condition 3.**  $\tilde{F}(\varphi, \lambda)$  is of class  $C^2$  with respect to  $\varphi$  for all  $\lambda \in \Lambda$  and both the derivatives are bounded uniformly with respect to  $\lambda$ .

To state the convergence result in Theorem 6.1 of [21], let  $\bar{t} > t_0$  be the largest value  $t$  such that the IVP (1.1) has a unique solution with initial data  $\varphi$  over the interval  $[t_0, \bar{t}]$  and denote the simulation mesh by  $\Delta = \{t_i\}_{i=1}^N$  with corresponding step size  $h_\Delta$ . Finally, let the approximation of the solution  $x_{t_n}$  obtained using a FCRK method with mesh  $\Delta$  given by  $\hat{\gamma}^n$ . Then, Maset et al. [21] prove

**Theorem 2.6** (Theorem 6.1 of [21]). *Assume that an FCRK method  $(A(\theta), b(\theta), c)$  of uniform order  $q$ , discrete order  $p$ , with  $p, q \in \{1, 2, 3, 4\}$ , and such that  $c_i \in [0, 1], i = 1, \dots, s$ , is applied to the distributed DDE (1.1) for the computation of  $x(t)$  through  $(t_0, \varphi) \in \mathbb{R} \times C_0([\tau_{max}, \tau_{min}])$ . Furthermore, assume that:*

- A:**  $\sum_{i=1}^s b_i(\theta) = \theta$  for  $\theta \in [0, 1]$  and  $\sum_{j=1}^s A_{i,j}(\theta) = \theta$  for all  $i$  with  $\theta \in [0, c_i]$ ;
- B:** Conditions 1, 2, and 3 hold;
- C:** The approximation error (2.7) satisfies  $\varepsilon = \mathcal{O}(h_\Delta^{\min(q+1,p)})$ ;
- D:**  $x(t)$  is five times continuously differentiable.

Then, for a fixed  $T \in [t_0, \bar{t}]$  and simulation meshes  $\Delta$  that include all possible discontinuity points of  $x$  in  $[t_0, \bar{t}]$ ,

$$\max_{t \in [t_0, T]} \|\hat{\gamma}^n - x_{t_n}\| = \mathcal{O}(h_\Delta^{\min(q+1,p)}).$$

We now apply the result of Theorem 2.6 to distributed DDEs with compact support. As before, we focus on the quadrature method and only consider existing FCRK methods that satisfy Assumption A. Furthermore, we assume that  $F$  is at least four times continuously differentiable, globally Lipschitz, and that  $F$  and its derivatives are bounded. Furthermore, we assume that all possible breaking points are contained in the simulation mesh  $\Delta$  and the solution  $x(t)$  is five times differentiable for  $t > t_0$ . We discuss breaking points of these distributed DDEs in Section 3. Thus, Assumption D is satisfied.

To show that  $\tilde{F}$  in Equation (2.6) satisfies the Conditions 1, 2, and 3, we consider arbitrary functions  $\varphi \in C_0([\tau_{max}, \tau_{min}])$  and quadrature rules with bounded weights

$$\sum_{i=1}^M |\sigma_i| < C_1 \tag{2.8}$$

for a fixed constant  $C_1$ . Now, for accurate quadrature methods and differentiable  $F$ , the argument in Equation (2.5) suggests that Condition 1 should hold. Indeed, the proof in the Appendix of [20] directly verifies this condition. Similarly, Condition 2 follows directly from the argument in [20], and both these arguments are similar to those in Section 6 of [21].

We must now show that  $\tilde{F}(\varphi, \lambda)$  is  $C^2$  with respect to  $\varphi$  for all  $\lambda$ . We recall that  $F$  is at least four times continuously differentiable and note that the integrand is linear in  $\varphi$ . Therefore, consecutive applications of the chain rule for Fréchet derivatives gives the

required regularity of  $\tilde{F}$ . Furthermore, the quadrature weights  $\sigma_i$  satisfy Equation (2.8) and we have assumed that  $F^{(k)}$  for  $k = 1, 2, 3, 4$  is bounded. Thus,  $\tilde{F}^{(l)}$  for  $l = 1, 2$  is uniformly bounded with respect to  $\lambda$  and we conclude that Condition 3, and therefore, Assumption B hold.

Consequently, it only remains to verify Assumption C. We consider composite quadrature rules  $\lambda$  of order  $q$  with maximal step size  $h_{int}$  to calculate  $\tilde{F}$ . Specifically, these quadrature rules satisfy

$$\sum_{i=1}^M \sigma_i x(t - \pi_i) k(\pi_i) - \int_{\tau_{min}}^{\tau_{max}} x(t - s) k(s) ds = \mathcal{O}(h_{int}^q).$$

Adding zero and Taylor expanding the latter expression in Equation (2.6) gives

$$\begin{aligned} F\left(x(t), \sum_{i=1}^M \sigma_i x(t - \pi_i) k(\pi_i)\right) &= F\left(x(t), \int_{\tau_{min}}^{\tau_{max}} x(t - s) k(s) ds\right. \\ &\quad \left.+ \left[\sum_{i=1}^M \sigma_i x(t - \pi_i) k(\pi_i) - \int_{\tau_{min}}^{\tau_{max}} x(t - s) k(s) ds\right]\right) \\ &= F\left(x(t), \int_{\tau_{min}}^{\tau_{max}} x(t - s) k(s) ds\right) \\ &\quad + F'\left(x, \int_{\tau_{min}}^{\tau_{max}} x(t - s) k(s) ds\right) \times h_{int}^q + \mathcal{O}(h_{int}^{2q_{int}}). \end{aligned}$$

The boundedness of  $F'(x, \int_{\tau_{min}}^{\tau_{max}} x(t - s) k(s) ds)$  gives  $\varepsilon_i(\varphi, \lambda) = \mathcal{O}(h_{int}^q)$ . In Section 2, we chose  $h_{int}$  such that  $\mathcal{O}(h_{int}^q) = \mathcal{O}(h_\Delta^p)$ . It follows that  $\varepsilon(\varphi, \lambda)$  satisfies Assumption C. We therefore conclude that Theorem 2.5 holds.

### 3 | Numerical Implementation and Convergence Examples

To implement the FCRK method described in the preceding sections, we must evaluate the quadrature integrals

$$\hat{I}(t) = \sum_{i=1}^M \sigma_i x(t - \pi_i) k(\pi_i).$$

Recalling that the kernel  $k$  and quadrature method  $(\sigma_i, \pi_i)$  are assumed to be known, we evaluate  $x(t - \pi_i)$  to approximate the distributed delay in Equation (1.1). As the quadrature method is fixed, the quadrature points  $\pi_i$  correspond to evaluating the solution  $x$  at a fixed time,  $t - \pi_i$  in the past. Consequently, the numerical method for the distributed DDE is equivalent to a numerical method for a discrete delay DDE with  $M$  delays corresponding to the collocation points of the quadrature method.

In what follows, we consider the composite Riemann, trapezoidal, and Simpson's quadrature methods with respective orders 1, 2, and 4. We will solve the numerically equivalent multidelay discrete DDE using the ddesd solver implemented in Matlab [23]. This solver is a continuous Runge–Kutta method, rather than an FCRK method. However, we do not consider distributed DDEs

with vanishing delays, so  $\tau_{min} > h$ , where  $h$  is the step size of the Runge–Kutta method, in any of our test problems. Therefore, there will be no overlapping and the continuous RK method behaves the same as the FCRK method [21]. Then, leveraging the discretization of the convolution integral in the FCRK method, we directly simulate the distributed DDE by

$$\left. \begin{aligned} \frac{d}{dt} u_{h_{int}}(t) &= F\left(u_{h_{int}}(t), \sum_{i=1}^M \sigma_i u_{h_{int}}(t - \pi_i) k(\pi_i)\right) \\ u_{h_{int}}(s) &= \varphi(s) \quad \text{for } s \in [-\tau_{max}, 0] \end{aligned} \right\} \quad (3.1)$$

where  $(\sigma_i, \pi_i)$  is a given quadrature method with step size  $h_{int}$  and we denote the explicit dependence of the solution on the step size by  $u_{h_{int}}(t)$ .

To evaluate the accuracy of the numerical scheme, we compare the numerical solution of Equation (3.1),  $u_{h_{int}}$ , against a reference solution of the distributed DDE Equation (1.1),  $x(t)$ . We evaluate the accuracy of the FCRK method by computing the  $L_\infty$  global error induced by the quadrature approximation with quadrature step size  $h_{int}$

$$E(h_{int}) = \max_{t \in [0, T_f]} |x(t) - u_{h_{int}}(t)|. \quad (3.2)$$

We obtain the reference solution  $x(t)$  by simulating systems of discrete DDEs that are equivalent to the distributed DDE (1.1) for specific choices of distribution kernel  $k(s)$ . We solve the equivalent system of DDEs using `ddesd` with relative and absolute error tolerances of  $10^{-12}$  solved over the interval  $t \in [0, t_F]$ . These equivalent discrete DDEs are typically obtained through variants of the linear chain technique [9, 20]. The linear chain technique, which first appeared in [27] and was popularized by MacDonald [28], consists of writing the convolution integral in Equation (1.1) as the solution of a system of auxiliary differential equations. The most used instance of the linear chain technique occurs in the reduction of an infinite delay distributed DDE to a system of transit compartment ODEs with possibly variable transit rates [29]. In recent work, Guglielmi and Hairer [30] used a variant of this idea and the fact that hypoexponential distributions are dense in the space of positively supported distributions to propose a numerical method for general distributed DDEs with infinite delays. A number of authors [31, 32] have studied the conditions on the  $k$  that permit this reduction to a system of ODEs.

There has been increased interest in developing equivalent systems of differential equations for distributed DDEs where the delay has compact support. For example, in the case of a uniform distribution over the interval  $[\tau_{min}, \tau_{max}]$ , it is well-known that the resulting distributed DDE is equivalent to a system of two discrete DDEs [14, 15]. Furthermore, if the kernel  $k$  is a polynomial function, such as in the case of a beta distribution, Pulch [13] recently derived an equivalent system of discrete DDEs. We leverage the ideas underlying the linear chain technique to derive a (to our knowledge) novel equivalent system of discrete DDEs in the case where the kernel  $k$  is a sum of exponential functions, which generalizes mixtures of exponential distributions. In each case, the equivalent discrete DDEs can be solved using existing techniques.

### 3.1 | Breaking Points

Solutions of DDEs typically have discontinuous derivatives at the initial time point  $t_0$ , as there is no guarantee that the history function  $\varphi$  satisfies

$$\varphi'(t_0^-) \neq F\left(\varphi(t_0), \int_0^\infty \varphi(t_0 - s) g_a^j(s) ds\right) = x'(t_0^+),$$

where the superscript represents limits from the left and right. These discontinuity points are called *breaking points* and typically propagate throughout higher-order derivatives of the solution through so-called smoothing. In our numerical simulations, we use the Matlab code `ddesd`, which does not explicitly detect breaking points, but rather implements residual error control via adaptive step sizes [24]. Thompson [33] remarked that this error control results in a method that roughly equivalent to a (3,4) Runge–Kutta pair. As breaking points that are not explicitly included in the simulation mesh typically result in a lower effective order of the numerical method, `ddesd` may exhibit third, rather than fourth, order convergence when solving DDEs with breaking points.

As mentioned, the solution of the distributed DDE (1.1)  $x$  is typically not continuously differentiable at  $t_0$ . Therefore, breaking points occur when the delayed argument satisfies  $t - \tau = t_0$  [17]. However, distributed DDEs typically benefit from additional smoothing due to the convolution integral Equation (2.4). We now show how this additional smoothing depends on the kernel  $k(s)$ . We consider the distributed DDE (1.1) and recall the convolution integral,  $I(t)$ , is given in Equation (2.4). As  $I(t)$  is the convolution of a continuous function,  $x$ , with the kernel  $k$ , it is continuous. As the majority of kernels,  $k$ , utilized in applications are differentiable,  $I(t)$  is typically also differentiable. We now proceed under the assumption that  $k$ , and thus  $I$ , are differentiable.

If  $I(t)$  is continuously differentiable at  $t = \tau_{min}$ , then the solution  $x$  will be twice continuously differentiable at  $t = \tau_{min}$  since  $F$  is assumed to be five times continuous differentiable. By rewriting  $I(t)$  as

$$I(t) = \int_{t-\tau_{max}}^{t-\tau_{min}} x(s)k(t-s)ds,$$

we calculate

$$\begin{aligned} \frac{d}{dt} I(t) &= x(t - \tau_{min})k(\tau_{min}) - x(t - \tau_{max})k(\tau_{max}) \\ &+ \int_{t-\tau_{max}}^{t-\tau_{min}} x(s)k'(t-s)ds. \end{aligned}$$

We expect a breaking point at  $t = t_0 + \tau_{min}$  and  $t = t_0 + \tau_{max}$  as  $x$  is not continuously differentiable at the initial time,  $t_0$ . However, if  $k(\tau_{min}) = k(\tau_{max}) = 0$ , then we find

$$\frac{d}{dt} I(t) = \int_{t-\tau_{max}}^{t-\tau_{min}} x(s)k'(t-s)ds,$$

so  $I'(t)$ , and thus  $x''(t)$ , is continuous. Then, integrating by parts and once again using  $k(\tau_{min}) = k(\tau_{max}) = 0$  gives

$$\frac{d}{dt}I(t) = \int_{t-\tau_{max}}^{t-\tau_{min}} x'(s)k(t-s)ds.$$

We can then repeat the same argument for higher-order derivatives of  $I$ . Therefore, if  $k(\tau_{min}) = k(\tau_{max}) = 0$ , then  $\tau_{min}$  and  $\tau_{max}$  are not breaking points of any order for the distributed DDE (1.1). We immediately see, that if  $k(\tau_{min}) \neq 0$  (or  $k(\tau_{max}) \neq 0$ ) then the distributed DDE will have a breaking point at  $t = t_0 + \tau_{min}$  (or  $t = t_0 + \tau_{max}$ ). We note that while the Matlab code `dde23` does allow for breaking point detection, it is inefficient for the relative and absolute error tolerances of  $1 \times 10^{-12}$  that we use in the following examples.

### 3.2 | Example Problems

For each of the three explicit distributions  $k$ , we consider two test problems solved over the interval  $t \in [0, 10]$ . The first is the linear distributed DDE

$$\left. \begin{aligned} \frac{d}{dt}x(t) &= \alpha x(t) + \beta \int_{\tau_{min}}^{\tau_{max}} x(t-s)k(s)ds \\ x(s) &= \varphi(s) \quad \text{for } s \in [-\tau_{max}, 0] \end{aligned} \right\} \quad (3.3)$$

with  $\alpha = -0.75, \beta = -1.25, \tau_{min} = 1.25, \tau_{max} = 2.95$ , and a constant history  $\varphi(s) = 1$ . The second test problem is a delayed logistic equation

$$\left. \begin{aligned} \frac{d}{dt}x(t) &= \alpha x(t) + \beta \left[ \int_{\tau_{min}}^{\tau_{max}} x(t-s)k(s)ds \right]^2 \\ x(s) &= \varphi(s) \quad \text{for } s \in [-\tau_{max}, 0], y(t) = \end{aligned} \right\} \quad (3.4)$$

with  $\alpha = 0.35, \beta = -0.25, \tau_{min} = 1.25, \tau_{max} = 2.95$ , and a constant history  $\varphi(s) = 1$ . In both cases, we observed similar convergence results hold for other parameterizations of both problems.

#### 3.2.1 | Uniform Distribution

We begin with the simplest example of a uniformly distributed delay with support in the interval  $[\tau_{min}, \tau_{max}]$ . In this case, the kernel  $k_{unif}(s)$  is given by

$$k_{unif}(s) = \begin{cases} \frac{1}{\tau_{max}-\tau_{min}} & \text{if } s \in [\tau_{min}, \tau_{max}] \\ 0 & \text{otherwise.} \end{cases} \quad (3.5)$$

As mentioned, the distributed DDE (1.1) with  $k(s) = k_{unif}(s)$  is equivalent to a system of two discrete DDEs [14, 15]

**Theorem 3.1** (Theorem 4.2 [15]). *The IVP (1.1) where  $k_{unif}(s)$  is the uniform distribution in Equation (3.5) is equivalent to the*

system of discrete DDEs

$$\left. \begin{aligned} \frac{d}{dt}x(t) &= F(x(t), y(t)) \\ \frac{d}{dt}y(t) &= \frac{1}{\tau_{max}-\tau_{min}} [x(t-\tau_{min}) - x(t-\tau_{max})] \end{aligned} \right\} \quad (3.6)$$

with initial data

$$\begin{aligned} x(s) &= \varphi(s) \quad \text{for } s \in [-\tau_{max}, 0] \quad \text{and} \\ y(0) &= \int_{\tau_{min}}^{\tau_{max}} \varphi(-s)k_{unif}(s)ds. \end{aligned}$$

We simulate the linear Equation (3.3) and nonlinear Equation (3.4) test problems using the equivalent system of two discrete DDEs in Equation (3.6) to obtain the reference solution  $x(t)$ . Then, for  $(\sigma_i, \pi_i)$  corresponding to the composite Riemann, trapezoidal, and Simpson quadrature methods with step size  $h_{int}$ , we simulate Equation (3.1) to obtain  $u_{h_{int}}(t)$  and evaluate the error  $E(h_{int})$  in Equation (3.2). In Figure 1A, B, we plot  $\log(E(h_{int}))$  as a function of the composite step size  $h_{int}$  to illustrate the error induced by approximating Equation (3.3) by the corresponding quadrature approximations. As expected, we see linear convergence on the log scale with slopes of  $q = 1$  and  $q = 2$  in the Riemann and trapezoidal approximations, respectively. The kernel  $k_{unif}$  is constant and nonzero over the interval  $[\tau_{min}, \tau_{max}]$ , so there are breaking points at both  $t - \tau_{min}$  and  $t - \tau_{max}$ . As `dde23` is roughly equivalent to the (3.4) Runge-Kutta pair due to the residual error control [33], the third-order convergence for Simpson's composite method is dominated by the error of the Runge-Kutta, rather than the quadrature, method. Consequently, this example illustrates the loss of accuracy of the numerical method due to the existence of breaking points.

#### 3.2.2 | Polynomial Kernel

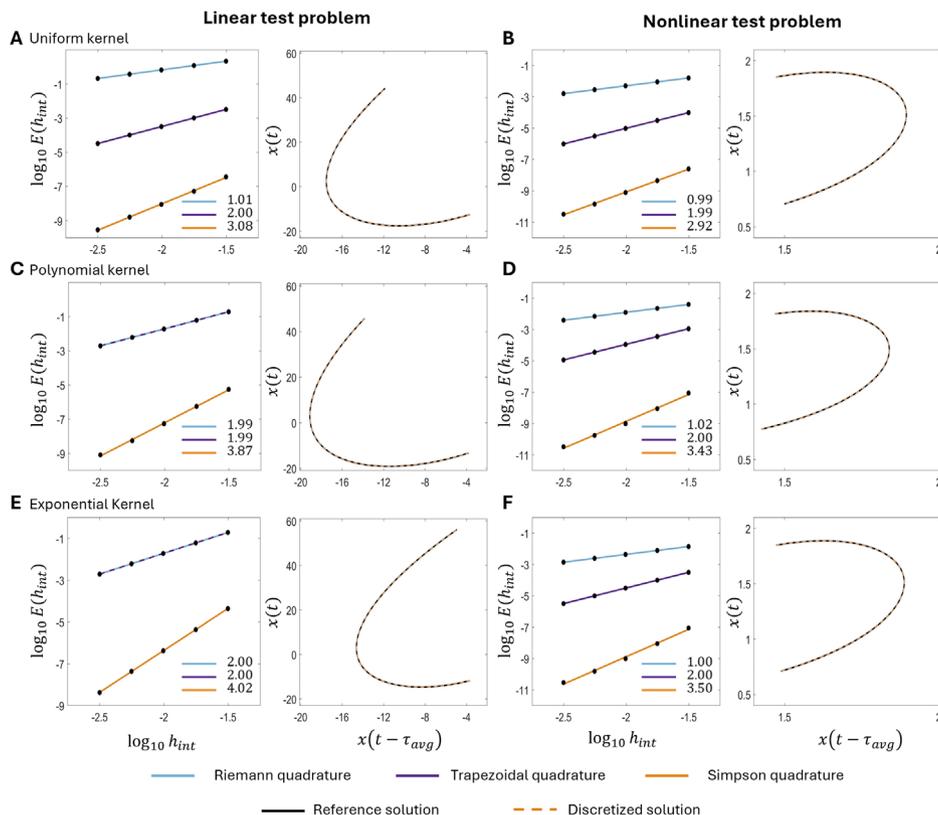
We next consider an  $n$ th degree polynomial kernel given by

$$k_{poly}(s) = \begin{cases} \sum_{i=0}^n a_i s^i & \text{if } s \in [\tau_{min}, \tau_{max}] \\ 0 & \text{otherwise} \end{cases} \quad (3.7)$$

subject to the conditions  $k_{poly}(s) \geq 0$  and  $\int_{\tau_{min}}^{\tau_{max}} k_{poly}(s)ds = 1$ . The corresponding distributed DDE is equivalent to the system of discrete DDEs [13].

**Theorem 3.2** (Theorem 2 [13]). *The IVP (1.1) where  $k_{poly}(s)$  is a polynomial distribution defined in Equation (3.7) is equivalent to the system of discrete DDEs*

$$\left. \begin{aligned} \frac{d}{dt}x(t) &= F(x(t), \sum_{i=0}^n a_i y_i(t)) \\ \frac{d}{dt}y_i(t) &= x(t-\tau_{min})(\tau_{min})^i - x(t-\tau_{max})(\tau_{max})^i - i y_{i-1}(t), \\ i &= 0, 1, 2, \dots, n \quad \text{with } y_{-1} = 0 \end{aligned} \right\} \quad (3.8)$$



**FIGURE 1** | Convergence of the FCRK method for distributed DDEs with compact support and comparison between the solutions  $x(t)$  and  $u_{h_{int}}$  for the linear and nonlinear test problems in Equations (3.3) and (3.4). The rows correspond to the uniform, polynomial, and exponential delay kernels, respectively. The first and third columns show the error  $\log_{10}(\max_{t \in [t_0, T]} |x(t) - u_{h_{int}}(t)|)$  as a function of the quadrature step size  $\log_{10}(h_{int})$  with  $u_{h_{int}}(t)$  obtained from Equation (3.3) with  $(\sigma_i, \pi_i)$  corresponding to the Riemann, trapezoidal, and Simpson’s quadrature methods. The slope of this curve gives the order of the corresponding numerical method and is reported in the legend for each quadrature method. The second and fourth columns show the reference solution  $x(t)$  plotted against  $x(t - \tau_{avg})$  in solid black and  $u_{h_{int}}(t)$  plotted against  $u_{h_{int}}(t - \tau_{avg})$  for  $u_{h_{int}}$  corresponding to the Simpson’s composite quadrature method in dashed orange, where  $\tau_{avg} = 0.5(\tau_{min} + \tau_{max})$ . The solutions are obtained using `ddestd` in Matlab with relative and absolute error tolerance of  $1 \times 10^{-12}$ .

with initial data

$$x(s) = \varphi(s) \text{ for } s \in [-\tau_{max}, 0] \text{ and } y_i(0) = \int_{\tau_{min}}^{\tau_{max}} \varphi(-s) s^i ds.$$

We consider the linear uniformly distributed DDE in Equation (3.3) with

$$k_{poly}(s) = \frac{(\tau_{min} - s)(\tau_{max} - s)}{\int_{\tau_{min}}^{\tau_{max}} (\tau_{min} - \theta)(\tau_{max} - \theta) d\theta}.$$

As before, we simulate the linear test problem Equation (3.3) using the equivalent system of discrete DDEs in Equation (3.8) and consider  $(\sigma_i, \pi_i)$  corresponding to the composite Riemann, trapezoidal, and Simpson quadrature methods with step size  $h_{int}$ , for  $u_{h_{int}}(t)$ . We note that  $k_{poly}(\tau_{min}) = k_{poly}(\tau_{max}) = 0$ , so the DDE has no breaking points and we expect `ddestd` to correspond to the underlying fourth-order method. In Figure 1C, corresponding to the linear test problem with a polynomial kernel, we plot  $\log(E(h_{int}))$  as a function of the composite quadrature step size  $h_{int}$ . The composite Riemann quadrature method only differs from the composite trapezoidal rule due to the inclusion of the end points at  $\tau_{min}$  and  $\tau_{max}$ . Therefore, these two quadrature methods are identical in the case where  $k_{poly}(\tau_{min}) =$

$k_{poly}(\tau_{max}) = 0$ . We therefore expect to gain an extra order of convergence in the Riemann method, as it is identical to the trapezoidal case in this case  $k_{poly}(\tau_{min}) = k_{poly}(\tau_{max}) = 0$ . Indeed, we see linear convergence on the log scale with slopes  $q = 2$  and  $q = 4$  for composite trapezoidal and Simpson approximations.

We next considered the kernel

$$\hat{k}_{poly}(s) = \frac{(0.75\tau_{min} - s)(\tau_{max} - s)}{\int_{\tau_{min}}^{\tau_{max}} (\tau_{min} - \theta)(\tau_{max} - \theta) d\theta}.$$

Here,  $\hat{k}_{poly}(\tau_{min}) \neq 0$ , so the distributed DDE has a breaking point at  $t = \tau_{min}$ . We simulate the nonlinear test problem Equation (3.4) for  $\hat{k}_{poly}$  using the equivalent system of discrete DDEs in Equation (3.8) to obtain the reference solution  $x(t)$ . For  $(\sigma_i, \pi_i)$  corresponding to the composite Riemann, trapezoidal, and Simpson quadrature methods with step size  $h_{int}$ , we simulate Equation (3.1) to obtain  $u_{h_{int}}(t)$ . In Figure 1D, corresponding to the nonlinear test problem, we plot  $\log(E(h_{int}))$  as a function of the composite step size  $h_{int}$ . As expected, we see linear convergence on the log scale with slopes of  $q = 1$  and  $q = 2$  in the Riemann and trapezoidal quadrature methods. The breaking point at  $t = \tau_{min}$  implies that the error of the Runge–Kutta method dominates the error of the composite Simpson’s quadrature method, which

results in the third-order convergence and a loss of accuracy of the method due to the presence of the breaking point.

### 3.2.3 | Exponential Kernel

Finally, we consider the kernel given by a sum of exponentially decaying functions restricted to the domain  $[\tau_{min}, \tau_{max}]$  and given by

$$k_{exp}(s) = \begin{cases} \sum_{i=0}^n a_i \exp(-\lambda_i s) & \text{if } s \in [\tau_{min}, \tau_{max}], \lambda_i \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.9)$$

subject to the conditions  $k_{exp}(s) \geq 0$  for all  $s$  and  $\int_{\tau_{min}}^{\tau_{max}} k(s)ds = 1$ . The kernel  $k_{exp}(s)$  is a mixture distribution of exponential distributions restricted to  $[\tau_{min}, \tau_{max}]$  if the coefficients  $a_i$  are a convex combination and  $\lambda_i > 0$ . Thus, Equation (3.9) can be considered a generalization of this class of distributions. Now, consider

$$A_i(t) = \int_{t-\tau_{max}}^{t-\tau_{min}} x(s) \exp(-\lambda_i(t-s))ds,$$

so that, after a change of variable,

$$\int_{\tau_{min}}^{\tau_{max}} k_{exp}(s)x(t-s)ds = \sum_{i=0}^n a_i A_i(t).$$

Now, as in the linear chain technique, we derive a system of auxiliary discrete DDEs for  $A_i(t)$ . Using Leibniz's rule, we differentiate  $A_i(t)$  with respect to  $t$  to find

$$\begin{aligned} \frac{d}{dt} A_i(t) &= x(t - \tau_{min}) \exp(-\lambda_i \tau_{min}) - x(t - \tau_{max}) \exp(-\lambda_i \tau_{max}) \\ &\quad - \lambda_i A_i(t) \quad \text{for } i = 1, \dots, n., \end{aligned}$$

with corresponding initial condition

$$A_i(0) = \int_{\tau_{min}}^{\tau_{max}} \varphi(t-s) \exp(-\lambda_i(s))ds.$$

Altogether, we obtain the equivalent system of discrete DDEs

$$\left. \begin{aligned} \frac{d}{dt} x(t) &= F\left(x(t), \sum_{i=0}^n a_i A_i(t)\right) \\ \frac{d}{dt} A_i(t) &= x(t - \tau_{min}) \exp(-\lambda_i \tau_{min}) - x(t - \tau_{max}) \exp(-\lambda_i \tau_{max}) \\ &\quad - \lambda_i A_i(t) \quad \text{for } i = 1, \dots, n \end{aligned} \right\} \quad (3.10)$$

For the linear test problem Equation (3.3), we consider the kernel

$$k_{exp}(s) = \frac{(0.25 - e^{-\lambda_1 s})(0.85 - e^{-\lambda_2 s})e^{-0.15s}}{\int_{\tau_{min}}^{\tau_{max}} (0.25 - e^{-\lambda_1 \theta})(0.85 - e^{-\lambda_2 \theta})e^{-0.15\theta} d\theta},$$

where  $\lambda_1 = -\log(0.25)/\tau_{min}$  and  $\lambda_2 = -\log(0.85)/\tau_{max}$  are chosen such that  $k_{exp}(\tau_{min}) = k_{exp}(\tau_{max}) = 0$  so the DDE has no breaking points. We simulate the linear test problem Equation (3.3) using the equivalent system of discrete DDEs in Equation (3.10) and consider  $(\sigma_i, \pi_i)$  corresponding to the composite Riemann, trapezoidal, and Simpson quadrature methods with step size  $h_{int}$ , for  $u_{h_{int}}(t)$ . In Figure 1E, corresponding to the linear test problem, we plot  $\log(E(h_{int}))$  as a function of the composite step size  $h_{int}$ . Again, the composite Riemann and trapezoidal quadrature methods are once again identical in this case. We see the expected linear convergence on the log scale with identical slopes of  $q = 2$  for these two methods, indicating a gain of accuracy for the Riemann method. The composite Simpson's rule displays the expected fourth-order convergence.

We next considered the kernel

$$\hat{k}_{exp}(s) = \frac{(0.25 - e^{-\lambda_1 s})(0.85 - e^{-\lambda_2 s})e^{-0.15s} + 0.02}{\int_{\tau_{min}}^{\tau_{max}} (0.25 - e^{-\lambda_1 \theta})(0.85 - e^{-\lambda_2 \theta})e^{-0.15\theta} + 0.02d\theta}.$$

Here, both  $\hat{k}_{exp}(\tau_{min}) \neq 0$  and  $\hat{k}_{exp}(\tau_{max}) \neq 0$ , so the distributed DDE has a breaking points at  $t = \tau_{min}$  and  $t = \tau_{max}$ . We simulate the nonlinear test problem Equation (3.4) for  $\hat{k}_{exp}$  using the equivalent system of discrete DDEs in Equation (3.8) to obtain the reference solution  $x(t)$ . For  $(\sigma_i, \pi_i)$  corresponding to the composite Riemann, trapezoidal, and Simpson quadrature methods with step size  $h_{int}$ , we simulate Equation (3.1) to obtain  $u_{h_{int}}(t)$ . In Figure 1F, corresponding to the nonlinear test problem, we plot  $\log(E(h_{int}))$  as a function of the composite step size  $h_{int}$ . As expected, we see linear convergence on the log scale with the expected slopes of  $q = 1, 2$  and  $q = 3$  for the composite Riemann, trapezoidal, and Simpson's quadrature methods.

## 4 | Complexity Collapse in the Mackey–Glass Equation

Tavakoli and Longtin [10] considered the Mackey–Glass equation [34] with multiple discrete delays given by

$$\frac{d}{dt} x(t) = -\gamma x(t) + \frac{\beta}{M} \sum_{i=1}^M \frac{x(t - \tau_i)}{1 + x(t - \tau_i)^{10}}, \quad (4.1)$$

where  $\{\tau_i\}_{i=1}^M$  is a bounded sequence of delays with mean  $\tau_{avg} = 17$ . The dynamics of the Mackey–Glass equation have been extensively well-studied as an example of a DDE that exhibits chaotic dynamics. Tavakoli and Longtin [10] considered  $\gamma = 1$  and  $\beta = 2$  and observed “complexity collapse” where the dynamics of Equation (4.1) simplify as the number of delays,  $M$ , increases. The authors noted the paradox present in this complexity collapse, as increasing the number of delays in a discrete DDE typically increases the complexity of the resulting dynamics. Here, we resolve this apparent paradox by showing that the multidelay discrete DDE is numerically indistinguishable from a distributed DDE, and demonstrate that this complexity collapse is a further example of distributed DDEs exhibiting less complex behavior than discrete DDEs.

To do so, we note that the sum of delays in Equation (4.1) is the Riemann sum discretization of the convolution integral

$$\frac{\beta}{M} \sum_{i=1}^M \frac{x(t - \tau_i)}{1 + x(t - \tau_i)^{10}} = \beta \frac{\tau_M - \tau_1}{M} \sum_{i=1}^M \frac{1}{\tau_M - \tau_1} \frac{x(t - \tau_i)}{1 + x(t - \tau_i)^{10}} \approx \beta \int_{\tau_1}^{\tau_M} \frac{x(t - s)}{1 + x(t - s)^{10}} k_{MG}(s) ds,$$

where  $k_{MG}$  is the uniform distribution over  $[\tau_1, \tau_M]$ . Our preceding analysis has focused on numerical methods for distributed DDEs where we discretize the convolution integral with an appropriately accurate quadrature rule and resulting multidelay discrete DDE and we have shown that these two representations are numerically indistinguishable. Of course, this relationship also works in the opposite direction, where the multidelay discrete DDE is indistinguishable from the distributed DDE: the multidelay discrete DDE in Equation (4.1) is the Riemann approximation of the uniformly distributed DDE

$$\frac{d}{dt} x(t) = -\gamma x(t) + \beta \int_{\tau_{min}}^{\tau_{max}} \frac{x(t - s)}{1 + x(t - s)^{10}} k_{MG}(s) ds, \quad (4.2)$$

where  $\tau_{max} - \tau_{min} = 1.75/4$ .

To simulate Equation (4.2), we use Leibniz's rule to write the convolution integral

$$I_{MG}(t) = \int_{\tau_{min}}^{\tau_{max}} \frac{x(t - s)}{1 + x(t - s)^{10}} k_{MG}(s) ds$$

as the solution of a discrete DDE via

$$\begin{aligned} \frac{d}{dt} I_{MG}(t) &= \frac{d}{dt} \int_{t-\tau_{max}}^{t-\tau_{min}} \frac{x(s)}{1 + x(s)^{10}} k_{MG}(t - s) ds \\ &= \frac{1}{\tau_{max} - \tau_{min}} \left( \frac{x(t - \tau_{min})}{1 + x(t - \tau_{min})^{10}} - \frac{x(t - \tau_{max})}{1 + x(t - \tau_{max})^{10}} \right). \end{aligned}$$

Thus, the distributed DDE (4.2) is equivalent to

$$\left. \begin{aligned} \frac{d}{dt} x(t) &= -\gamma x(t) + \beta I_{MG}(t) \\ \frac{d}{dt} I_{MG}(t) &= \frac{1}{\tau_{max} - \tau_{min}} \left( \frac{x(t - \tau_{min})}{1 + x(t - \tau_{min})^{10}} - \frac{x(t - \tau_{max})}{1 + x(t - \tau_{max})^{10}} \right) \end{aligned} \right\} \quad (4.3)$$

with initial data

$$\begin{aligned} x(s) &= \varphi(s) \quad \text{for } s \in [-\tau_{max}, 0] \quad \text{and} \\ I_{MG}(0) &= \int_{\tau_{min}}^{\tau_{max}} \frac{\varphi(-s)}{1 + \varphi(-s)^{10}} ds. \end{aligned}$$

This example differs from the linear and nonlinear examples considered in Section 3 as the integrand is nonlinear in the solution  $x(t)$ . We once again simulate the equivalent distributed DDE (4.3) using `ddesd` to obtain a reference solution. We simulate the corresponding discretization of the distributed DDE (4.2) using the Riemann, trapezoidal, and Simpson discretizations of

the convolution integral  $I_{MG}$ . In the prior examples, we used a constant history function. Here, we consider a nonconstant history function  $\varphi$  defined by the chaotic solution of the Mackey–Glass equation with a single discrete delay. Specifically, we defined the history function  $\varphi$  by the solution of

$$\left. \begin{aligned} \frac{d}{dt} \varphi(s) &= -\gamma \varphi(s) + \beta \frac{\varphi(s - \tau_{avg})}{1 + \varphi(s - \tau_{avg})^{10}} \\ \varphi(\xi) &= 1 \quad \text{for } \xi \in [-\tau_{avg}, 0] \end{aligned} \right\} \quad (4.4)$$

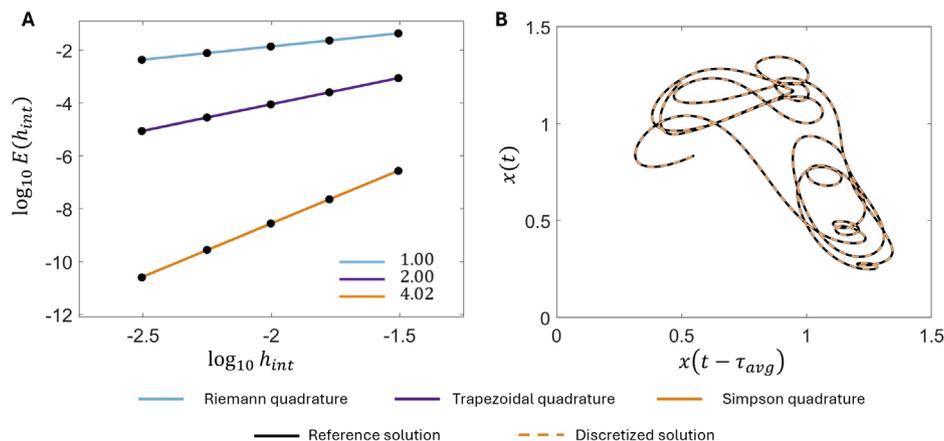
evaluated for  $s \in (85 - \tau_{max}, 85)$ . The auxiliary initial condition in Equation (4.3) is

$$I_{MG}(0) = \int_{\tau_{min}}^{\tau_{max}} \frac{\varphi(-s)}{1 + \varphi(-s)^{10}} k_{MG}(s) ds.$$

In Figure 2, we show the convergence of the multidelay DDEs corresponding to the discretization of the convolution integral to the true underlying solution  $x(t)$ . As the uniform distribution does not vanish at  $\tau_{min}$  or  $\tau_{max}$ , we expect the existence of breaking points with corresponding loss of accuracy. However, we observe linear convergence of the discretized DDE on the log scale with the slopes of  $q = 1, 2$  and  $q = 3$  for the composite Riemann, trapezoidal, and Simpson's quadrature methods. In this example, it appears that the residual error control implemented in `ddesd` is able to handle the breaking point with this nonconstant history function. We recall that the Riemann quadrature method corresponds precisely to the multidelay DDE considered by Tavakoli and Longtin [10]. We therefore conclude that the multidelay complexity collapse observed by Tavakoli and Longtin [10] is due to convergence of the multidelay DDE (4.1) to the distributed DDE (4.2) and thus an example of distributed DDEs having simpler dynamics than discrete delay DDEs, as has been observed previously [35].

## 5 | Discussion

In this work, we have utilized the existing framework of FCRK methods to illustrate how multidelay discrete DDEs can be used to accurately simulate compactly supported distributed DDEs. This numerical method relies on combining the convergence framework established by Maset et al. [21] with existing quadrature methods to discretize the convolution integral in Equation (1.1). We established the convergence of this numerical method by explicitly relating the quadrature method with the Runge–Kutta method by formalizing the intuition that the least accurate of the quadrature or Runge–Kutta method dominates the error. In this sense, this result is unsurprising. However, our results justify the common approach to simulating these distributed DDE in mathematical models that consists of discretizing the convolution integral and simulating the resulting multidelay discrete DDE. As this discretization, which can be delicate, corresponds to a choice of quadrature method, this work gives an explicit condition for the number of composite intervals needed to maintain the overall accuracy of the resulting numerical method. As such, models with distributed delays can be simulated by leveraging existing codes *without* requiring the development of any problem-specific software and our results



**FIGURE 2** | Convergence of the FCRK method and comparison between the solutions  $x(t)$  and  $u_{h_{int}}$  of the uniformly distributed Mackey–Glass equation in Equation (4.2). Panel A shows the error  $\log_{10}(\max_{t \in [t_0, T]} |x(t) - u_{h_{int}}(t)|)$  as a function of the quadrature step size  $\log_{10}(h_{int})$  with  $u_{h_{int}}(t)$  obtained from Equation (3.3) with  $(\sigma_i, \pi_i)$  corresponding to the Riemann, trapezoidal, and Simpson’s quadrature methods. The slope of this curve gives the order of the corresponding numerical method and is reported in the legend for each quadrature method. Panel B shows the reference solution  $x(t)$  plotted against  $x(t - \tau_{avg})$  in solid black and  $u_{h_{int}}(t)$  plotted against  $u_{h_{int}}(t - \tau_{avg})$  for  $u_{h_{int}}$  corresponding to the Simpson’s composite quadrature method in dashed orange, where  $\tau_{avg} = 0.5(\tau_{min} + \tau_{max})$ . The solutions are obtained using `ddesd` in Matlab with relative and absolute error tolerance of  $1 \times 10^{-12}$ .

may simplify the discretization of distributed DDEs commonly done during numerical bifurcation studies.

We illustrate our theoretical results through linear and nonlinear test problems for three specific types of distributions that each admit equivalent representations as systems of discrete DDEs. Two of these equivalent representations were known prior to this work. However, the equivalence between a distributed DDE where the distribution is a mixture of exponential distributions with a system of discrete DDEs had not been previously demonstrated. For these test problems, we used existing numerical methods for discrete DDEs to simulate the equivalent formulation and obtain a reference solution which we then compared against the solution obtained using the FCRK method.

In these numerical tests, we used the Matlab solver `ddesd`. This solver uses residual error control to adapt the step size of the numerical method rather than explicitly detecting breaking points. This is an important limitation, as our convergence framework explicitly assumes that all the breaking points of the DDE are included in the mesh. However, we established sufficient conditions on the kernel  $k$  to ensure that the resulting distributed DDE does not have breaking points. In the case where the distributed DDE does not have breaking points, we observed the expected convergence rates of the resulting FCRK method, including an additional order of accuracy for the Riemann method. Conversely, in the case where the distributed DDE has breaking points, our numerical tests nevertheless indicate convergence of the FCRK method, however at a lower order than the FCRK predicts, but consistent with the approximate order of `ddesd` [24].

Finally, we used the numerical equivalence between the distributed DDE and the multidelay discrete DDE to understand the complexity collapse observed by Tavakoli and Longtin [10]. Our analysis demonstrates that the multidelay DDEs considered in that work are approximations of the underlying uniformly

distributed DDE. Consequently, our results link the observed complexity collapse observed for discrete DDEs with many delays with the common intuition of simpler dynamics in distributed DDEs. Understanding how the bifurcation structure leading to chaotic behavior in these multidelay discrete DDEs, such as those found in Tavakoli and Longtin [36], depends on the quadrature method is an interesting avenue for future investigation.

Altogether, we have formalized the common approach to simulating distributed DDEs with compact support by using the convergence framework of [21] to demonstrate the convergence of the approximated distributed DDEs to the underlying model. We showed how to utilize efficient numerical solvers to simulate these distributed DDEs and gave a precise relationship between the accuracy of the quadrature method and the underlying Runge–Kutta method. Our results will therefore allow modelers to a priori choose the discretization parameters when simulating a distributed DDE and facilitate the use of more biologically realistic distributed DDEs throughout mathematical biology.

#### Data Availability Statement

The data that support the findings of this study are openly available in Compact\_Distributed\_DDEs at [https://github.com/ttcassid/Compact\\_Distributed\\_DDEs](https://github.com/ttcassid/Compact_Distributed_DDEs).

#### References

1. T. Cassidy, A. R. Humphries, M. Craig, and M. C. Mackey, “Characterizing Chemotherapy-Induced Neutropenia and Monocytopenia Through Mathematical Modelling,” *Bulletin of Mathematical Biology* 82 (2020): 104.
2. Y. M. Song, S. Campbell, L. Shiao, J. K. Kim, and W. Ott, “Noisy Delay Denoises Biochemical Oscillators,” *Physical Review Letters* 132 (2024): 78402–78409.
3. M. Craig, A. R. Humphries, and M. C. Mackey, “A Mathematical Model of Granulopoiesis Incorporating the Negative Feedback Dynamics and

- Kinetics of G-CSF/Neutrophil Binding and Internalization,” *Bulletin of Mathematical Biology* 78 (2016): 2304–2357.
4. H. Hong, M. J. Cortez, Y.-Y. Cheng, et al., “Inferring Delays in Partially Observed Gene Regulation Processes,” *Bioinformatics* 39 (2023): 1–11.
  5. A. Sargood, E. A. Gaffney, and A. L. Krause, “Fixed and Distributed Gene Expression Time Delays in Reaction – Diffusion Systems,” *Bulletin of Mathematical Biology* 84 (2022): 1–29.
  6. D. W. Kim, H. Hong, and J. K. Kim, “Systematic Inference Identifies A Major Source of Heterogeneity in Cell Signaling Dynamics: The Rate-Limiting Step Number,” *Science Advances* 8 (2022): 1–11.
  7. H. Smith, *An Introduction to Delay Differential Equations with Applications to the Life Sciences*, Vol. 57 of Texts in Applied Mathematics, ed. J. E. Marsden, L. Sirovich, and S. S. Antman (Springer New York, 2011).
  8. P. Hurtado and C. Richards, “Time is of the Essence: Incorporating Phase-Type Distributed Delays and Dwell Times Into ODE Models,” *Applied Mathematics in Science and Engineering* 1 (2020): 410–422.
  9. T. Cassidy, “Distributed Delay Differential Equation Representations of Cyclic Differential Equations,” *SIAM Journal on Applied Mathematics* 81 (2021): 1742–1766.
  10. S. K. Tavakoli and A. Longtin, “Multi-Delay Complexity Collapse,” *Physical Review Research* 2 (2020): 1–7.
  11. Z. Wang and A. R. Humphries, “Numerical Methods for Delay Differential Equations with Threshold State-Dependent Delay,” *IFAC-PapersOnLine* 55 (2022): 163–168.
  12. T. Gedeon, A. R. Humphries, M. C. Mackey, H.-O. Walther, and Z. Wang, “Operon Dynamics With State Dependent Transcription and/or Translation Delays,” *Journal of Mathematical Biology* 84 (2021): 1–74.
  13. R. Pulch, “Equivalent Systems for Differential Equations with Polynomially Distributed Delay,” ArXiv e-prints, 2408.09861, 2024, 1–15.
  14. A. Teslya, “Predator-Prey Models With Distributed Time Delay” (Doctoral diss., McMaster University, 2015).
  15. T. Cassidy, M. Craig, and A. R. Humphries, “Equivalences Between Age Structured Models and State Dependent Distributed Delay Differential Equations,” *Mathematical Biosciences and Engineering* 16 (2019): 5419–5450.
  16. L. Tavernini, “One-Step Methods for the Numerical Solution of Volterra Functional Differential Equations,” *SIAM Journal of Numerical Analysis* 8 (1971): 786–795.
  17. A. Bellen, S. Maset, M. Zennaro, and N. Guglielmi, “Recent Trends in the Numerical Solution of Retarded Functional Differential Equations,” *Acta Numerica* 18 (2009): 1–110.
  18. A. S. Eremin, “Runge-Kutta Methods for Differential Equations With Distributed Delays,” *AIP Conference Proceedings* 2116 (2019b): 140003.
  19. G. P. Langlois, M. Craig, A. R. Humphries, et al., “Normal and Pathological Dynamics of Platelets in Humans,” *Journal of Mathematical Biology* 75 (2017): 1411–1462.
  20. T. Cassidy, P. Gillich, A. R. Humphries, and C. H. van Dorp, “Numerical Methods and Hypoexponential Approximations for Gamma Distributed Delay Differential Equations,” *IMA Journal of Applied Mathematics* 87 (2022): 1043–1089.
  21. S. Maset, L. Torelli, and R. Vermiglio, “Runge-Kutta Methods for Retarded Functional Differential Equations,” *Mathematical Models and Methods in Applied Sciences* 15 (2005): 1203–1251.
  22. A. Eremin and A. Humphries, “Efficient Accurate Non-Iterative Breaking Point Detection and Computation for State-Dependent Delay Differential Equations,” *AIP Conference Proceedings* 1648 (2015): 150006.
  23. MATLAB, R2017a. 2017.
  24. L. Shampine, “Solving ODEs and DDEs with Residual Control,” *Applied Numerical Mathematics* 52 (2005): 113–127.
  25. A. S. Eremin, “Functional Continuous Runge-Kutta Methods With Reuse,” *Applied Numerical Mathematics* 146 (2019a): 165–181.
  26. A. Bellen and M. Zennaro, *Numerical Methods for Delay Differential Equations. Numerical Mathematics and Scientific Computation* (Oxford University Press, 2003).
  27. T. Vogel, “Systèmes Déferlants, Systèmes Héritaires, Systèmes Dynamiques,” in *Proceedings of International Symposium on Nonlinear Vibrations* (1961), 123–130.
  28. N. MacDonald, *Time Lags in Biological Models*, Vol. 27 of Lecture Notes in Biomathematics, ed. S. Levin (Springer, 1978).
  29. D. Câmara De Souza, M. Craig, T. Cassidy, et al., “Transit and Lifespan in Neutrophil Production: Implications for Drug Intervention,” *Journal of Pharmacokinetics and Pharmacodynamics* 45 (2018): 59–77.
  30. N. Guglielmi and E. Hairer, “Applying Stiff Integrators for ODEs and DDEs to Problems With Distributed Delays,” *SIAM Journal on Scientific Computing* 47, no. 1 (2025): A102–A123.
  31. O. Diekmann, M. Gyllenberg, and J. A. J. Metz, “Finite Dimensional State Representation of Linear and Nonlinear Delay Systems,” *Journal of Dynamics and Differential Equations* 30 (2018): 1439–1467.
  32. O. Diekmann, M. Gyllenberg, and J. A. J. Metz, “Finite Dimensional State Representation of Physiologically Structured Populations,” *Journal of Mathematical Biology* 80 (2020): 205–273.
  33. S. Thompson and L. Shampine, “A Friendly Fortran DDE Solver,” *Applied Numerical Mathematics* 56 (2006): 503–516.
  34. M. C. Mackey and L. Glass, “Oscillation and Chaos in Physiological Control Systems,” *Science* 197 (1977): 287–289.
  35. A. Thiel, H. Schwegler, and C. W. Eurich, “Complex Dynamics is Abolished in Delayed Recurrent Systems With Distributed Feedback Times,” *Complexity* 8 (2003): 102–108.
  36. S. K. Tavakoli and A. Longtin, “Dynamical Invariants and Inverse Period-doubling Cascades in Multi-Delay Systems,” *Chaos: An Interdisciplinary Journal of Nonlinear Science* 31 (2021): 103129.