QMC Methods for the solution of delay differential equations ⋆

Reinhold Kainhofer

Department of Mathematics, Graz University of Technology, Steyrergasse 30, A-8010 Graz, Austria

Abstract

In this paper the Quasi-Monte Carlo methods for Runge Kutta solution techniques of differential equations, which were developed by Stengle, Lécot, Coulibaly and Koudiraty, are extended to delay differential equations of the form $f(t, y(t), y(t - \tau(t)))$. The retarded arguments are approximated by interpolation, after which the conventional (Quasi-)Monte Carlo Runge Kutta methods can be applied. We give a proof of the convergence of this method and its order in a general form, which does not depend on a specific Quasi-Monte Carlo Runge Kutta method. Finally, a numerical investigation shows that - as with ordinary differential equations - for heavily oscillating delay differential equations, this quasi-randomized method leads to an improvement compared even to high order Runge Kutta schemes.

Key words: Delay differential equation, Quasi-Monte Carlo methods, Runge-Kutta methods

1 Introduction

In physics and other engineering subjects, the rate of change of a process $y(t)$ often does not only depend on the value of $y(t)$ at time t, but also on the values of the process in the past. In the general form, the "delay differential equation" for the process then can be written in the form:

$$y'(t) = f(t, y(t), y(t - \tau_1(t)), ...y(t - \tau_k(t))) \quad \text{for } t > t_0, \tag{1}$$
$$y(t) = y_0(t) \quad \text{for } t \leq t_0 \, . \tag{2}$$

⋆ This research was supported by the Austrian Science Foundation Project S-8308-MAT

The most noticeable difference to ordinary initial value problems is that the initial value must be given as a function on a certain interval (which is basically $\left[\inf_{\substack{t\geq t_0 \\ 0\leq j\leq k}} t-\tau_j(t), t_0\right]$) instead of only the value at the starting point t_0.

In [1] and [2] Stengle proposed a (randomized) Monte Carlo algorithm for the solution of the initial value problem

$$y'(t) = f(t, y(t)), \quad 0 < t < T, \qquad y(0) = y_0 \ , \tag{3}$$

where f is smooth in y but only bounded and Borel measurable in t. The family of solution methods he proposed is akin to the Runge-Kutta family, and thus called the Runge-Kutta Monte Carlo (RKMC) family. In his derivation he does not discretize the time, but only the spatial dimensions y, and solves the remaining integral equation using Monte Carlo integration. Lécot [3], Coulibaly and Lécot [4] and Lécot and Koudiraty [5] generalized this method for orders up to 3, and additionally used Quasi-Monte Carlo methods to calculate the resulting integral (thus the name RKQMC methods for them).

The application of these Monte Carlo and lately also Quasi-Monte Carlo methods to numerical integration and integral equations has been extensively studied in literature. While Monte Carlo methods employ random points to estimate a given integral, Quasi-Monte Carlo methods lay more emphasis on good distribution rather than randomness - quite often they are referred to as deterministic variants of Monte Carlo method. Instead of using (pseudo) random numbers, in Quasi-Monte Carlo methods deterministic sequences are used which have a very good uniform distribution, usually called low-discrepancy sequences. The uniformity of distribution of a point set S consisting of N points $x_0, \ldots, x_{N-1}$ is measured in terms of its discrepancy, defined as

$$D_N(S) = \sup_{a,b\in[0,1]^s} \left| \frac{A([a,b), S)}{N} - \lambda_s([a,b)) \right|, \tag{4}$$

where $A(E, S)$ denotes the number of points $x_j \in S$, $0 \leq j < N$ that lie inside the interval E, and $\lambda_s(E)$ is the s-dimensional Lebesgue measure. The most prominent sequences with the lowest known orders of discrepancy $D_N(S) \leq \mathcal{O}\left(\frac{(logN)^{s-1}}{N}\right)$, are Halton's sequences and so called (t, m, s) net sequences, with Sobol's, Faure's and Niederreiter's sequences as special cases. For a detailed discussion of these sequences and discrepancy in general, we refer the interested reader to the monographs of Niederreiter [6] and Drmota and Tichy [7].

In their papers, Stengle, Coulibaly, Koudiraty and Lécot apply the Runge-Kutta (Quasi) Monte Carlo methods only to ordinary differential equation with initial conditions as given in equation (3). We will first give a brief out-

line of their methods here, before applying them to delay differential equations. Under the assumptions mentioned above (f does not even have to be differentiable in t, let alone smooth), $f(y, y(t))$ is Taylor-expanded only with respect to $y(t)$ and the differential equation is recursively substituted into itself. This leads to an equation of the form (see [8] or [5]):

$$f(t_0 + h) = y(t_0) + \sum_{i=1}^{s} \int_{t_0}^{u_0} \cdots \int_{t_0}^{u_{i-1}} F_i \left(u_1, \ldots, u_i; y(t_0)\right) du_i \ldots du_1 + \mathcal{O}(h^{s+1}) \ , \quad (5)$$

where $u_0 = t_0 + h$ and the F_k, $1 \leq k \leq s$, are defined recursively by $F_0(y) = y$, $F_i := D_y F_{i-1} f(u_i, y)$. The sum is then combined into one s-dimensional integral over a new function G_s with $\bar{u} = \left(u_{\pi(1)}, \ldots, u_{\pi(s)}\right)$ such that $u_{\pi(1)} \leq u_{\pi(2)} \leq \cdots \leq u_{\pi(s)}$:

$$f(t_0 + h) = y(t_0) + \frac{1}{s! h_n^{s-1}} \int_{(t_0, t_0+h)^s} G_s \left(\bar{u}; y(t_0)\right) du + \mathcal{O}(h^{s+1}) \ , \quad (6)$$

The function G_s is rewritten using a suitable identity (and according order conditions on the coefficients) to get rid of the derivatives of F_i with respect to $y(t)$. Finally, the remaining integral is approximated by Monte Carlo integration. The s-order RKQMC method generates a sequence $(y_n)_{0 \leq n \leq n^*}$ defined by (see e.g. [3] or [5]):

$$y_{n+1} = y_n + \frac{h_n}{s! N} \sum_{0 \leq j < N} G_s \left(t_{n,j}, y(t_n)\right) \ , \quad (7)$$

where $t_{n,j} = t_n + h_n \bar{x}_j$ and $\bar{x}_j^{(1)} \leq \bar{x}_j^{(2)} \leq \cdots \leq \bar{x}_j^{(s)}$ are the elements of the s-dimensional uniformly distributed sequence $(x_j)_{0 \leq j \leq N}$ with their dimensions sorted in ascending order. The explicit forms of the functions G_1, G_2 and G_3 as given by Lécot [3] and Koudiraty [5] are shown in equations (15) to (17), the corresponding order conditions can be found in the cited papers (esp. [8, equations (4.12) to (4.15)]).

We will use this method and extend it to delay differential equations in the rest of the paper. Our discussion will not depend on one specific RKQMC method, however, we will also give specific proofs for the convergence and its order of the RKQMC1 [3], RKQMC2 [3] and RKQMC3 [8] methods applied to delay differential equations.

2 Description of the problem

In this paper, we consider initial value problems for delay (also called retarded) differential equations (DDE) with one retarded argument having the form

$$\begin{aligned} y'(t) &= f\left(t, y(t), y(t-\tau(t))\right), \quad \text{for } t \geq t_0, \\ y(t) &= \phi(t), \ \text{for } t \leq t_0 \ , \end{aligned} \tag{8}$$

where $y(t)$ is a d-dimensional real-valued function (which is in general not smooth), $\tau(t)$ is the continuous delay function, which we assume to be bounded from below ($0 < \tau_0 = \inf_{t \geq t_0} \tau(t)$). Furthermore, $\phi(t)$ is the initial function, which is assumed to be piecewise continuous at least on the interval $(\inf_{t_0 \leq t}(t - \tau(t)), t_0)$. We will not discuss the simplest case of a constant delay function $\tau(t) = \tau_0$, but instead the more general case of a delay function $\tau(t)$ satisfying

$$t_1 - \tau(t_1) \leq t_2 - \tau(t_2) \quad \text{for } t_1 \leq t_2 \ , \tag{9}$$

i.e. $T(t) := t - \tau(t)$ is an increasing function of t.

Solutions of the differential equation (8) are continuous for all $t > t_0$ and piecewise differentiable. Provided that $f(t, y, z)$ is (locally) Lipschitz w.r.t. y and z, the existence and uniqueness theorems for ordinary differential equations carry over verbatim. However, even if f and ϕ are smooth, the solution $y(t)$ is only smooth if $\phi(t)$ solves the differential equation (8). Otherwise, the solution $y(t)$ will have discontinuous derivatives $y^{(j)}(t_k)$ for $j \geq k$ at times t_k, which are recursively defined by $t_k = T^{-1}(t_{k-1})$ (see [9]). This means that with each additional time interval of length $h_k = t_k - t_{k-1}$, the discontinuities are smoothed out, and all the derivatives up to the k-th are continuous. For a constant delay function, we have $T^{-1}(t_{k-1}) = t_{k-1} + \tau_0$, and the solution $y(t)$ has k continuous derivatives at $t_k = t_0 + kau_0$, and $y^{(k+1)}(t)$ in general has a jump discontinuity at t_k. If the initial function $\phi(t)$ (or its derivatives) have discontinuities, a similar statement holds with t_0 replaced by the discontinuities of the initial function (see [9]).

3 Runge Kutta QMC Methods applied to delay differential equations

We will first discuss how the delayed argument $y(t - \tau(t))$ can be treated, and then deduce the Runge Kutta Quasi-Monte Carlo scheme for delay differential equations, which is a straight generalization of Stengle's, Lécot's and Koudiraty's schemes applied to this type of differential equations. However,

due to the special nature of delay differential equations, some things need to be handled more carefully.

3.1 Treatment of the retarded value

The main obstacle when solving delay differential equations is the way to treat the additional argument $y(t-\tau(t))$ which depends on the past values of the solution. Several attempts have been made (mainly by Bellman [10] et al.) to calculate the retarded values by recursively using the DDE itself, however, with this approach the number of calculations increase drastically with T. The other approach to the retarded argument is interpolation (see e.g. [11]), which we will use. Since for the numerical solution of the differential equation, the time is split into discrete time steps, labeled with index n, we have the approximated values of the solution at the discrete times $t_j (0 \leq j < n)$, available when calculating $y_n = y(t_n)$ for $n \geq 0$. It seems natural to use an interpolating function $P_q(t; (y_i)_{i \leq n})$ to approximate the values $y(t-\tau(t))$ from the values $y(t_j)$, $0 \leq j < n$, if $t - \tau(t) \geq t_0$. If $t - \tau(t) < t_0$, the initial function $\phi(t)$ can be used for the exact values of the solution, anyway.

Since the solutions of the differential equation are not smooth, the interpolation must be done using only past values with $\bar{\tau}_1 \leq t_k \leq \cdots \leq t - \tau(t) \leq \cdots \leq t_{k+m} \leq \bar{\tau}_2$ where $(\bar{\tau}_1, \bar{\tau}_2)$ is the largest interval of smoothness containing $t - \tau(t)$, i.e. $\bar{\tau}_1$ and $\bar{\tau}_2$ are the closest points where the derivatives of the solution have discontinuities as discussed in the last section. As the value of the derivative of $y(t)$ can be easily calculated for each time t_j by simply inserting into the differential equation, Hermite interpolation (as investigated in [9]) is used here since it only requires half the points ordinary Lagrange interpolation would require. Although the interpolating function then also depends on the approximated values of the derivative of $y(t)$, we will suppress this in our notation.

Note that the requirement of the interpolation points lying inside an interval of smoothness also poses a practical restriction on the value of the delay function:

$$\tau(t) \geq ph \qquad \text{or} \qquad h \leq \frac{\tau(t)}{p}\,, \tag{10}$$

where p is the number of points needed for interpolation, and h is the time step, if the time is discretized into fixed time intervals. Note also, that this restriction is sufficient, but not the best possible, and of course it can also be seen as an upper bound for the time step h.

If stepsize control methods are applied to the Runge-Kutta method (i.e. the time step h_j of the j-th step is not fixed but chosen such that the error does

not exceed a prescribed threshold), the restriction reads $\tau(t) \geq \sum_{j=0}^{p-1} h_{\nu-j}$ where ν is the largest number for which $t_\nu = t_0 + \sum_{j=1}^{\nu} h_j < t$, assuming a smooth initial function. (If the initial function is not smooth, the additional discontinuities of the derivatives force an even tighter bound.)

If these restrictions are not fulfilled, one can still try to obtain a RKQMC scheme by a Taylor-Expansion in the retarded argument as well, although this would complicate the algorithm and its derivation considerably.

These restrictions are the reason why mostly low-order methods for delay differential equations have been considered. In the sequel, we will also not develop high order methods, but instead look at the effect of the application of Quasi-Monte Carlo methods to Runge Kutta algorithms for delay differential equations.

By approximating the retarded value by the past values of the solution, we can directly insert this interpolation function

$$z(t) = z_{(y_i)_{i\leq n}}(t) = \begin{cases} \phi(t-\tau(t)), & \text{if } t-\tau(t) \leq t_0 \\ P_q\left(t-\tau(t); (y_i), (y_i')\right) & \text{otherwise} \end{cases} \tag{11}$$

into the differential equation and arrive at an ordinary differential equation

$$y'(t) = f(t, y(t), y(t-\tau(t))) \approx f(t, y(t), z(t)) =: g(t, y(t)) \; . \tag{12}$$

Applying the RKQMC methods for ordinary differential equations (equation (7) or [4,3,8,5]), we can approximate a solution of the delay differential equation.

3.2 The RKQMC schemes for DDE

In the sequel, we will use a notation which is independent of the Runge Kutta scheme, and only where necessary use scheme-specific results. We will furthermore assume $h_n < 1$ for all n.

Using the notation of [8], the exact value $y(t_{n+1})$ is approximated by the s-order Runge-Kutta (Q)MC scheme as:

$$y(t_{n+1}) = y(t_n) + \frac{1}{s! h_n^{s-1}} \int_{(t_n, t_{n+1})^s} G_s\left(\bar{u}; y(t_n); y(t \leq t_n)\right) du + h_n \varepsilon_n \tag{13}$$

where G_s is the differential increment function of the scheme, and ε_n is the local truncation error. This integral is then approximated by Quasi-Monte

Carlo integration and interpolation of the retarded argument as

$$y_{n+1} = y_n + \frac{h_n}{s!N} \sum_{0 \le j < N} \tilde{G}_s \left(\bar{t}_{j,n}; y_n, (y_i)_{i \le n} \right), \tag{14}$$

where $\tilde{G}_s(t; y; (y_i)_{i \le n}) := G_s \left(t; y; z_{(y_i)_{i \le n}}(\cdot) \right)$ denotes the function G_s with the retarded values interpolated from the points $(y_i)_{i \le n}$.

For the first-, second- and third-order RKQMC schemes of Lécot [3] and Koudiraty [8], the corresponding functions G_s are ([3,8]):

$$G_1(u; y; z(\cdot)) = f(u, y, z(u - \tau(u))) \tag{15}$$

$$\begin{aligned} G_2(\bar{u}; y; z(\cdot)) &= f(\bar{u}_1, y, z(\bar{u}_1 - \tau(\bar{u}_1))) + \frac{1}{\beta} f(\bar{u}_2, y, z(\bar{u}_2 - \tau(\bar{u}_2))) + \\ &+ \frac{1}{\alpha} f(\bar{u}_2, y + \alpha h_n f(\bar{u}_1, y, \bar{u}_1 - \tau(\bar{u}_1)), z(\bar{u}_2 - \tau(\bar{u}_2))) \end{aligned} \tag{16}$$

$$\begin{aligned} G_3(\bar{u}; y; z(\cdot)) &= a_1 f(\bar{u}_1, y, z(\bar{u}_1 - \tau(\bar{u}_1))) + \\ &+ \sum_{l=1}^{L_2} a_{2,l} f(\bar{u}_2, y + b_{2,l} h_n f(\bar{u}_1, y, z(\bar{u}_1 - \tau(\bar{u}_1))), z(\bar{u}_1 - \tau(\bar{u}_1))) + \\ &+ \sum_{l=1}^{L_3} a_{3,l} \Big(\bar{u}_3, y + b_{3,l}^{(1)} h_n f(\bar{u}_1, y, z(\bar{u}_1 - \tau(\bar{u}_1))) + \\ &\quad + b_{3,l}^{(2)} h_n f(\bar{u}_2, y_n + c_{3,l} h_n(\bar{u}_1, y_n), z(\bar{u}_2 - \tau(\bar{u}_2))), z(\bar{u}_1 - \tau(\bar{u}_1)) \Big) \end{aligned} \tag{17}$$

The condition on α and β is $\frac{1}{\alpha} + \frac{1}{\beta} = 1$ [3], and the conditions for the coefficients in G_3 can be found in [8].

3.3 Convergence proof

In the convergence proof we will use the following definitions for the Runge Kutta error δ_n, the QMC approximation error d_n, the interpolation error η_n and the local truncation error ε_n:

$$\delta_n := \frac{1}{s! h_n^s} \int_{(t_n, t_{n+1})^s} \left\{ \tilde{G}_s \left(\bar{u}; y_n, (y_i)_{i \le n} \right) - \tilde{G}_s \left(\bar{u}; y(t_n), (y_i)_{i \le n} \right) \right\} du \tag{18}$$

$$d_n := \frac{1}{s!N} \sum_{0 \le j < N} \tilde{G}_s \left(\bar{t}_{j,n}; y_n, (y_i)_{i \le n} \right) - \frac{1}{s! h_n^s} \int_{(t_n, t_{n+1})^s} \tilde{G}_s \left(\bar{u}; y_n, (y_i)_{i \le n} \right) du \tag{19}$$

$$\eta_n := \frac{1}{s! h_n^s} \int_{(t_n, t_{n+1})^s} \left\{ \tilde{G}_s \left(\bar{u}; y(t_n), (y_i)_{i \le n} \right) - G_s \left(\bar{u}; y(t_n), y(t \le t_n) \right) \right\} du \tag{20}$$

$$\varepsilon_n := \frac{y(t_{n+1}) - y(t_n)}{h_n} - \frac{1}{s! h_n^s} \int_{(t_n, t_{n+1})^s} G_s \left(\bar{u}; y(t_n); y(t \le t_n) \right) du \tag{21}$$

Since the interpolation does not explicitly depend on the RKQMC scheme used, we can give a general bound for the interpolation error η_n:

Proposition 1 *If G_s is Lipschitz continuous in its second and third arguments (with Lipschitz constant $\mathcal{L}_2$), the interpolation is chosen such that the interpolation error is of order p, and the interpolation is Lipschitz continuous in the sense that $\tilde{G}_s$ fulfills*

$$\left\|\tilde{G}_s\left(t, y(t); (v_i, v_i')_{i\leq n}\right) - \tilde{G}_s\left(t, y(t); (w_i, w_i')_{i\leq n}\right)\right\| \leq \tilde{\mathcal{L}}_2 \max_{i\leq n} \|v_i - w_i\| \quad (22)$$

for $v_i, w_i \in B(y(t_i), \rho)$, $\rho > 0$ and $t \leq t_i + h_i$ for all $i \leq n$, then the interpolation error η_n is bounded by

$$\|\eta_n\| \leq \frac{\mathcal{L}_2'}{s!}\left(\max_{j\leq n}\|e_j\| + Kh_n^p\right) \quad (23)$$

with constants $\mathcal{L}_2' = \max(\mathcal{L}_2, \tilde{\mathcal{L}}_2)$ and $K > 0$.

Remark 2 *Without loss of generality, we do not need to include the values of the derivatives at the right hand side of assumption (22), since they are calculated by the delay differential equation itself. The Lipschitz-continuity of G_s in general demands the Lipschitz-continuity of $f(t, y(t), y(t-\tau(t)))$, so that the difference in the derivatives can be bounded by the difference of the function values using this Lipschitz-continuity.*

PROOF. Adding and subtracting the term $G_s(\bar{u}, y(t_n); z_{(y(t_i))_{i\leq n}}(\cdot))$ to the integrand of the definition of η_n and using the Lipschitz conditions on G_s and $\tilde{G}_s$ yields the result

$$\begin{aligned}\|\eta_n\| \leq \frac{1}{s!h_n^s}\Bigg(&\tilde{\mathcal{L}}_2 \max_{i\leq n}\|y_i - y(t_i)\|\, h_n^s + \\ &+\mathcal{L}_2 \int_{(t_n, t_{n+1})^s} \max_{1\leq j\leq s}\left\|z_{(y(t_i))_{i\leq n}}(u_j - \tau(u_j)) - y(u_j - \tau(u_j))\right\| du\Bigg) \leq \\ &\leq \frac{\mathcal{L}_2'}{s!}\left(\max_{j\leq n}\|e_n\| + Kh^p\right) .\end{aligned}$$

Theorem 3 *Let G_s be Lipschitz continuous in its second and third arguments with Lipschitz constant $\mathcal{L}_2$ and of bounded variation in the sense of Hardy and Krause, the RKQMC method be chosen such that for an order $p \geq 1$ there exist $c_1(h) = \mathcal{O}(1)$, $c_2(h) = \mathcal{O}(1)$ and $c_3(h) = \mathcal{O}(1)$ with*

$$\|\varepsilon_n\| \leq c_1(h_n)h_n^p \quad (24)$$
$$\|\delta_n\| \leq c_2(h_n)\,\|e_n\| \quad (25)$$
$$\|d_n\| \leq c_3(h_n)D_N^*(S)\ , \quad (26)$$

where $S = (\mathbf{x}_1, \dots, \mathbf{x}_N)$ is the point set used for the QMC integration. If the retarded values are interpolated by a q-order method, and the assumptions of Proposition 1 are fulfilled, then the error $\|e_{n+1}\|$ of the corresponding RKQMC method for delay differential equations is bounded by

$$\|e_n\| \leq \|e_0\| \, e^{t_n\left(c_2 + \frac{\mathcal{L}_2}{s!}\right)} + \frac{e^{t_n\left(c_2+\frac{\mathcal{L}_2}{s!}\right)} - 1}{c_2 + \frac{\mathcal{L}_2}{s!}} \left\{ c_3 D_N^*(X) + \frac{\mathcal{L}_2}{s!} M H^q + c_1 H^p \right\} , \tag{27}$$

where $H = \max_{0\leq i\leq n-1} h_i$ and $c_i = \max_{0\leq j\leq n-1} c_i(h_j) = c_i(H)$.

Remark 4 *If the RKQMC method used is convergent for ordinary differential equations, the existence of $c_1(h_n)$, $c_2(h_n)$ and $c_3(h_n)$ is ensured since this is the most important part of the convergence proof for ODE. For this reason, we will not try and give the assumptions of theorem 3 in a more fundamental way. Inequality* (24) *states that the RK method is at least of order p and inequality* (26) *is basically a consequence of f and thus G_s being of bounded variation in the sense of Hardy and Krause, so that the Quasi-Monte Carlo integration error can be estimated by the famous Koksma-Hlawka inequality.*

Proof of theorem 3: Combining the methods of [8] and [9], the error $e_n = y_n - y(t_n)$ of the scheme can be written as

$$\begin{aligned}
e_{n+1} = y_{n+1} - y(t_{n+1}) &= \{y_n - y(t_n)\} + \\
&+ \frac{h_n}{s!} \left\{ \frac{1}{N} \sum_{0\leq j<N} \tilde{G}_s\left(\bar{t}_{j,n}; y_n, (y_i)_{i\leq n}\right) - \frac{1}{h_n^s} \int_{(t_n,t_{n+1})^s} \tilde{G}_s\left(\bar{u}; y_n, (y_i)_{i\leq n}\right) du \right\} + \\
&+ \frac{1}{s! h_n^{s-1}} \int_{(t_n,t_{n+1})^s} \left\{ \tilde{G}_s\left(\bar{u}; y_n, (y_i)_{i\leq n}\right) - \tilde{G}_s\left(\bar{u}; y(t_n), (y_i)_{i\leq n}\right) \right\} du + \\
&+ \frac{1}{s! h_n^{s-1}} \int_{(t_n,t_{n+1})^s} \left\{ \tilde{G}_s\left(\bar{u}; y(t_n), (y_i)_{i\leq n}\right) - G_s\left(\bar{u}; y(t_n), y(t \leq t_n)\right) \right\} du - \\
&- h_n \varepsilon_n \, .
\end{aligned}$$

And so

$$\begin{aligned}
\|e_{n+1}\| \leq \|e_{n+1}\|_{\text{bnd}} &:= \|e_n\| + h_n \|d_n\| + h_n \|\delta_n\| + h_n \|\eta_n\| + h_n \|\varepsilon_n\| \\
&\leq \|e_n\| + h_n c_3(h_n) D_N^*(X) + h_n c_2(h_n) \|e_n\| + c_1(h_n) h_n^{p+1} + \\
&\quad + h_n \frac{L_2}{s!} \left\{ M h_n^q + \max_{j\leq n} \|e_n\| \right\} \\
&\leq \left(1 + h_n \left(c_2 + \frac{\mathcal{L}_2}{s!}\right)\right) \|e_n\|_{\text{bnd}} + h_n \left(c_3 D_N^*(X) + \frac{\mathcal{L}_2}{s!} M H^q + c_1 H^p \right)
\end{aligned}$$

Recursively inserting this inequality into itself, using the inequality $1 + h_n a \leq e^{h_n a}$ and a telescopic sum finally yield the result.

Remark 5 *One should note that in the proof given in [9], it is assumed that the approximation formula $z(t)$ to the retarded values is smooth, and so the set of support points for the interpolation can only be changed between two time steps. Since the QMC integration needs the calculation of a lot of points from a large interval, a significant amount of extrapolation would be involved. However, the assumptions of the RK(Q)MC methods by Stengle, Lécot, Coulibaly and Koudiraty, and the method presented here do not require the function $g(t, y(t))$ to be smooth in t any more, but demand only boundedness of g and its derivatives w.r.t. y for all t. This is still fulfilled if we choose the support points different for different values of t (but always choose the same support points in a finite neighborhood of t), thus avoiding unnecessary extrapolation.*

3.4 Convergence of the RKQMC1, RKQMC2 and RKQMC3 methods

For the first- to third-order RKQMC methods, Lécot and Koudiraty proved the following lemmas under the assumptions that

(1) there exist $t > 0, \rho > 0$ such that for $0 \leq m \leq s$, $D_y^m f$ is measurable on the set $E := \cup_{0 \leq t \leq T} [t, \min(t+\tau, T)] \times B(y(t), \rho)$ and, for fixed t, continuous in y on the open ball $B(y(t), \rho)$
(2) and that for every $t \in [0, T]$ and every $y \in B(y(t), \rho)$ the m-th derivative $D_y^m f(u, y)$ is defined for $u \in [t, \min(t+\tau, t)]$ and bounded by $\left\| D_y^m f \right\|_E$ for $0 \leq m \leq s$, and its variation is bounded by $V_E(D_y^m f)$ for $0 \leq m \leq s-1$.

Lemma 6 (Lécot, [3]) *For the RKQMC1 method, if $h_n \leq \tau$ and $\|e_n\| + h_n \|f\|_{\infty,\varepsilon} < \rho$, then*

$$\|\varepsilon_n\| \leq \frac{h_n}{2} \|f\|_{\infty,\varepsilon} \left\| D_y^1 f \right\|_{\infty,\varepsilon} \tag{28}$$

$$\|d_n\| \leq \left\| D_y^1 f \right\|_{\infty,\varepsilon} \|e_n\| \tag{29}$$

$$\|\delta_n\| \leq V_\varepsilon(f) D_N^*(X) \tag{30}$$

Lemma 7 (Lécot, [3]) *If for the RKQMC2 method the additional conditions $h_n \leq \tau$ and $\|e_n\| + (1+\alpha) h_n \|f\|_{\infty,\varepsilon} < \rho$ hold, then with $D_i = \left\| D_y^i f \right\|_{\infty,\varepsilon}$, $i = 1, 2$*

the following inequalities hold:

$$\|\varepsilon_n\| \leq \frac{h_n^2}{12} \|f\|_{\infty,\varepsilon} \left((3\alpha^2 + 2) \|f\|_{\infty,\varepsilon} D_2 + 2(D_1)^2 \right) =: c_1^{(2)}(f) h_n^2 \tag{31}$$

$$\|d_n\| \leq \frac{1}{2} \left(1 + \frac{1}{|\alpha|} + \frac{1}{|\beta|} + h_n D_1 \right) D_1 \|e_n\| =: c_2^{(2)}(f, h_n) \|e_n\| \tag{32}$$

$$\|\delta_n\| \leq \frac{1}{2} \left(1 + \frac{1}{|\alpha|} + \frac{1}{|\beta|} + h_n \left(D_1 + 2V_\varepsilon(D_y^1 f) \right) \right) V_\varepsilon(f) D_N^*(X) =: \tag{33}$$

$$=: c_3^{(2)}(f, h_n) D_N^*(X) \tag{34}$$

Lemma 8 (Koudiraty, [8]) *If $h_n \leq \tau$ and $\|e_n\| + h_n c^* \|f\|_E < \rho$ with*

$$c^* := 1 + \max \left(\max_{1 \leq l \leq L_1} |b_{2,l}|, \max_{1 \leq l \leq L_3} \left| b_{3,l}^{(1)} \right| + \max_{1 \leq l \leq L_3} \left| b_{3,l}^{(2)} \right|, \max_{1 \leq l \leq L_3} |c_{3,l}| \right)$$

hold for the RKQMC3 method, then there exist $c_1(h_n) = \mathcal{O}(1)$, $c_2(h_n) = \mathcal{O}(1)$ and $c_3(h_n) = \mathcal{O}(1)$ such that

$$\|\varepsilon_n\| \leq c_1(h_n) h_n^3 \tag{35}$$

$$\|\delta_n\| \leq c_2(h_n) \|e_n\| \tag{36}$$

$$\|d_n\| \leq c_3(h_n) D_N^*(X) \tag{37}$$

Inserting these lemmas in our result from theorem 3, one obtains the convergence proofs for the RKQMC1, RKQMC2 and RKQMC3 methods for delay differential equations:

Corollary 9 *For the first order RKQMC1 method applied to delay differential equations, if the past values are interpolated in a way that the assumptions of Proposition 1 are fulfilled, and $H \leq \tau$ as well as*

$$e^{\left\|D_y^1 f\right\|_{\infty,\varepsilon} t_n} \|e_0\| + \frac{e^{Dt_n} - 1}{D} \left(\frac{1}{2} \|f\|_{\infty,\varepsilon} DH + V_\varepsilon(f) D_N^*(X) \right) + H \|f\|_{\infty,\varepsilon} < \rho$$

hold with $D = \left\|D_y^1 f\right\|_{\infty,\varepsilon}$, then the method is convergent for $h_n \to 0$, $D_N^*(X) \to 0$ and $\|e_0\| \to 0$, and the error is bounded by

$$\|e_n\| \leq \|e_0\| e^{(D+\mathcal{L}_2)t_n} + \frac{e^{(D+\mathcal{L}_2)t_n} - 1}{D + \mathcal{L}_2} \left(V_\varepsilon(f) D_N^*(X) + \mathcal{L}_2 M H^q + \frac{1}{2} \|f\|_{\infty,\varepsilon} DH \right) \tag{38}$$

Corollary 10 *For the second order RKQMC2 method for delay differential equations, the error under the assumptions of Lemma 7 and proposition 1 is bounded by* (27) *with $c_1 := c_1^{(2)}(f)$, $c_2 := \max_{0 \leq j < n} c_2^{(2)}(f, h_n)$ and $c_3 := \max_{0 \leq j < n} c_3^{(2)}(f, h_n)$ with the definitions from equations* (31) *to* (33)*, and thus*

the method is convergent. Its order is $\min(2,q)$*, if a low-discrepancy point set is used with* $N = \mathcal{O}\left(H^{-\min(2,q)}\right)$.

Corollary 11 *For the third order RKQMC3 method for delay differential equations, the error under the assumptions of Lemma 8 and proposition 1 is bounded by* (27) *with* $c_1^{(3)}(h_n)$*,* $c_2^{(3)}(h_n)$ *and* $c_3^{(3)}(h_n)$ *as defined in [8]. So the method is convergent with order* $\min(3,q)$*, if a low-discrepancy point set is used with* $N = \mathcal{O}\left(H^{-\min(3,q)}\right)$

4 Computational experiments

We now investigate the effects of the use of Runge Kutta QMC methods compared to conventional Runge Kutta methods at the example of the delay differential equation

$$y'(t) = 3y(t-1)sin(\lambda t), \qquad \text{for } t \geq 0 \tag{39}$$
$$y(t) = 1, \qquad \text{for } t \leq 0 \tag{40}$$

with $\lambda = 2^\nu$ and $1 \leq \nu \leq 16$. The exact solutions for different values of ν is shown in figure 1. As one can see, the solution oscillate heavily, although their amplitude get smaller as ν grows. One also has to notice the kink at $t = 1$ for all solutions, which is the first discontinuity of $y'(t)$ due to the initial function not being a solution of the DDE.

In the sequel, if not mentioned explicitly, all calculations were done with a time step of $h_n = h = 0.001$ for values up to $T = 5$, and the past values are interpolated by a fourth order Hermite interpolation. To compare the RKQMC methods with conventional Runge Kutta methods, we use the RKQMC methods presented in the previous chapter, as well as some low order Runge Kutta Methods (the 4-stage Runge method of order 3 and the 3-stage method of Heun of order 3). Additionally, we computed the results with Butcher's 6-th order method as an example of a high order Runge Kutta scheme. These schemes are discussed in [12] at great detail, so we will not even give the Butcher tableaus here.

To compare the various methods, we compute the average error

$$err = \frac{1}{K}\sum_{i=1}^{K} |y(t_{im}) - y_{im}|$$

of a method where K and m are chosen such that $t_{im} - t_{(i-1)m} = 0.1$ and $t_{Km} = T$.

We are interested in the convergence order of the RKQMC methods applied

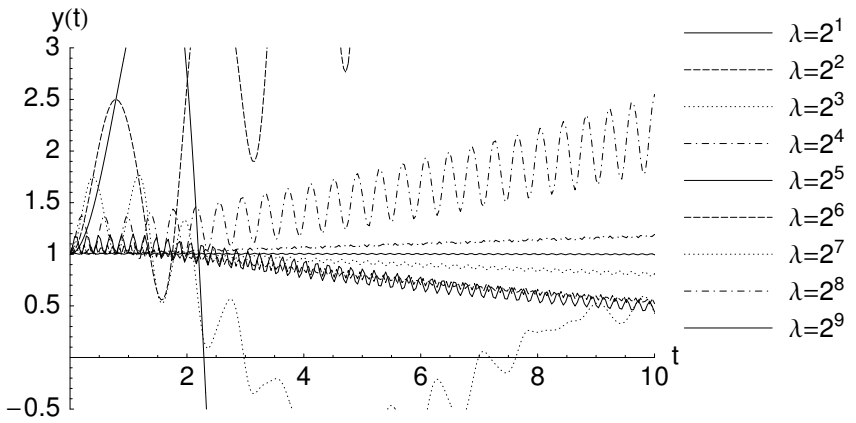

FIGURE 1. THE EXACT SOLUTION OF (39) FOR SOME VALUES OF λ.

to delay differential equations. In theorem 3 and its corollaries 9, 10 and 11, the error bound depends on the discrepancy of the point set used for the simulation, as well as on the method itself, and on h_n. If we take $N = \mathcal{O}(h_n^i)$, $i = 1, 2, 3$, according to the corollaries we would expect a different convergence order for the RKQMC1, RKQMC2 and RKQMC3 method. For $\lambda = 2^5$, however, we do not observe any difference with this specific DDE, neither for different orders of N, nor for the different RKQMC schemes, as figure 3 shows (other delay differential equations show the expected behavior).

A very important question when dealing with QMC methods is how many points to use for the integration. Taking too many points does not hurt accuracy, but unnecessarily wastes computing time. On the other hand, if one takes N too little, the Quasi-Monte Carlo integration error will not become negligible and thus introduce a bias to the result. If we take a fixed $h_n = 0.001$, $\lambda = 2^5$ and just increase the number of points, figure 2 shows that values of N larger than about $2^9 = 512$, cannot noticeably improve the error any more, because then the Runge Kutta and the truncation error outweigh the Quasi-Monte Carlo error. One can also notice that with low-discrepancy sequences like Sobol's sequence, which we used for all our calculations, the curve is much smoother than with pseudo-random numbers as used in Monte Carlo methods. This of course is caused by the good distribution properties of these sequences, and can improve the error considerably below the Monte Carlo error (see e.g. [8]).

As Stengle [2] pointed out, the RK(Q)MC methods need a lot more evaluations of f, and thus can only claim advantage if these calculations can be carried out in parallel. Additionally, the RKQMC methods might even outperform conventional high order Runge-Kutta schemes, or calculations with a much lower h_n. The results of Stengle, Lécot and Koudiraty show that, their RK(Q)MC methods only gain advantage for differential equations where f varies significantly faster in t than in space. Figure 4 and table 1 show the

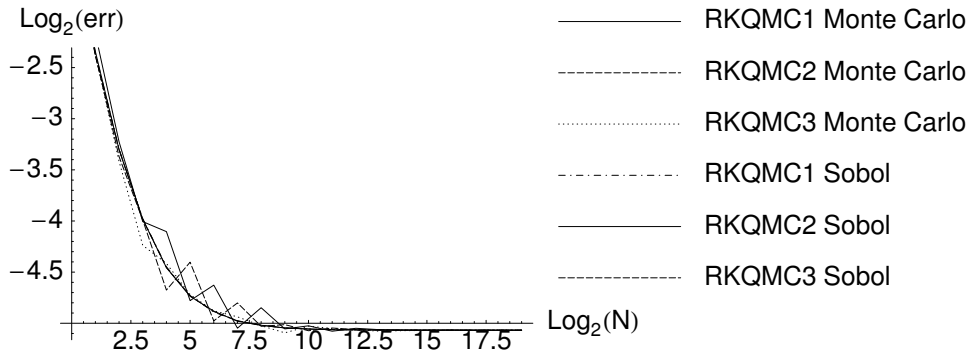

FIGURE 2. WITH FIXED h_n AND INCREASING N, THERE IS NO VISIBLE DIFFERENCE IN THE CONVERGENCE ORDER ($\lambda = 32$).

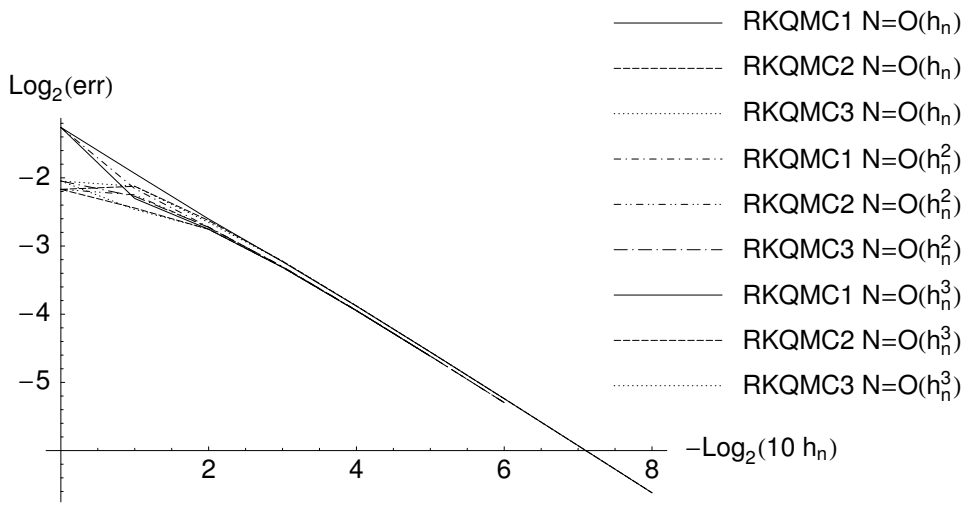

FIGURE 3. ERROR FOR $N = \mathcal{O}(h_n)$, $N = \mathcal{O}(h_n^2)$ AND $N = \mathcal{O}(h_n^3)$. ($\lambda = 32$)

error for the various methods for different values of ν in $\lambda = 2^\nu$. The time steps are $h_n = 0.001$ for the Butcher, Runge and Heun method, and $h_n = 0.01$ for the RKQMC methods. While for small values of λ the conventional Runge Kutta methods clearly give better results than the randomized Runge Kutta schemes, for high values of λ the situation changes, and the RKQMC schemes (which are low order methods) at least reach the same or a better average error than even the high order Butcher method.

5 Conclusion

In this paper we showed that the randomized Runge Kutta schemes as proposed by Stengle, Lécot, Coulibaly and Koudiraty can be successfully extended

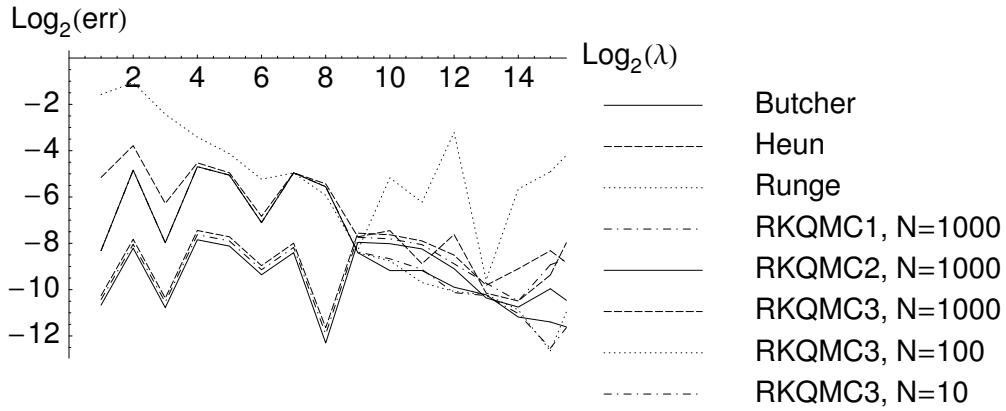

FIGURE 4. FOR RAPIDLY VARYING DDE ($\lambda \geq 2^{10}$), RKQMC METHODS OUTPERFORM CONVENTIONAL HIGH ORDER RUNGE KUTTA SCHEMES.

λ	Butcher	Runge	RKQMC1 $N = 1000$	RKQMC2 $N = 1000$	RKQMC3 $N = 1000$	RKQMC3 $N = 10$
2^1	−10.6663	−10.4277	−8.31395	−8.31394	−8.31389	−1.57259
2^2	−8.20716	−8.04143	−4.83919	−4.83909	−4.83908	−1.0597
2^3	−10.7767	−10.5397	−7.97728	−7.97651	−7.97628	−2.42371
2^4	−7.84711	−7.62639	−4.69098	−4.69087	−4.69086	−3.41549
2^5	−8.11869	−7.89766	−5.05594	−5.05595	−5.05624	−4.13132
2^6	−9.35768	−9.14174	−7.10261	−7.10411	−7.10611	−5.23092
2^7	−8.40319	−8.17748	−4.9544	−4.95436	−4.95707	−4.95889
2^8	−12.303	−11.9188	−5.54122	−5.55782	−5.55972	−5.92666
2^9	−7.95694	−7.73868	−8.4076	−8.37118	−8.40286	−8.30661
2^{10}	−8.0165	−7.80652	−8.75515	−8.66781	−9.17772	−5.1714
2^{11}	−8.24265	−8.05737	−9.69635	−9.12556	−9.17449	−6.22673
2^{12}	−9.09544	−8.88462	−10.0774	−10.1313	−9.89475	−3.20923
2^{13}	−10.3594	−9.72112	−10.2139	−10.2555	−10.2228	−9.58793
2^{14}	−10.7523	−10.4912	−10.8732	−11.0333	−11.1829	−5.65672
2^{15}	−9.96253	−8.92665	−12.6424	−12.5296	−11.396	−4.90893
2^{16}	−10.956	−8.4328	−9.28126	−10.7504	−11.8211	−3.4884
2^{17}	−8.85172	−9.77356	−9.94702	−10.4492	−9.48664	−5.50948
2^{18}	−8.80146	−8.58722	−11.3812	−10.3786	−10.1079	−6.2911

TABLE 1. ERROR FOR INCREASING PARAMETER VALUES λ ($h_n = 0.001$ FOR BUTCHER, RUNGE AND HEUN, AND $h_n = 0.01$ FOR RKQMC)

to delay differential equations, and for rapidly varying differential equations, even the low order RKQMC methods lead to an improvement over conventional (high- and low-order) Runge Kutta schemes. Although the RKQMC methods need several times more function evaluations than its Runge Kutta counterparts of the same order, the possibility of parallel computing and the fact that the RKQMC methods even outperform high order Runge Kutta schemes for certain types of delay differential equations, make the RKQMC methods for delay differential equations a viable solution method.

References

[1] G. Stengle, Numerical methods for systems with measureable coefficients, Appl. Math. Lett. 3 (1990) 25–29.

[2] G. Stengle, Error analysis of a randomized numerical method, Numer. Math. 70 (1995) 119–128.

[3] C. Lécot, Quasi-randomized numerical methods for systems with coefficients of bounded variation, Mathematics and Computers in Simulation 55 (2001) 113–121.

[4] I. Coulibaly, C. Lécot, A Quasi-randomized Runge-Kutta Method, Mathematics of Computation 68 (226) (1999) 651–659.

[5] C. Lécot, A. Koudiraty, Numerical Analysis of Runge-Kutta Quasi-Monte Carlo Methods, submitted (2001).

[6] H. Niederreiter, Random Number Generation and Quasi-Monte Carlo Methods, Society for industrial and applied mathematics, Philadelphia, Pennsylvania, 1992.

[7] M. Drmota, R. Tichy, Sequences, Discrepancies and Applications, Vol. 1651 of Lecture Notes in Mathematics, Springer, New York, Berlin, Heidelberg, Tokyo, 1997.

[8] A. A. Koudiraty, C. Lécot, Numerical analysis of some quasi-Monte Carlo Methods, submitted (2001).

[9] H. J. Oberle, H. J. Pesch, Numerical Treatment of Delay Differential Equations by Hermite Interpolation, Numer. Math. 37 (1981) 235–255.

[10] R. Bellman, On the Computational Solution of Differential-Difference Equations, J. Math. Anal. and Appl. 2 (1961) 108–110.

[11] J. Oppelstrup, The RKFHB4 Method for Delay-Differential Equations, in: R. D. G. Roland Burlirsch, J. Schröder (Eds.), Numerical Treatment of Differential Equations, Vol. 631 of Lecture Notes in Mathematics, Springer, New York, Berlin, Heidelberg, Tokyo, 1976, pp. 133–146, proceedings, Oberwolfach, 1976.

[12] E. Hairer, S. P. Norsett, G. Wanner, Solving Ordinary Differential Equations I, Vol. 8 of Springer Series in Computational Mathematics, Springer, Berlin, Heidelberg, 1987.