QMC METHODS FOR THE SOLUTION OF DIFFERENTIAL EQUATIONS WITH MULTIPLE DELAYED ARGUMENTS*

Reinhold Kainhofer and Robert F. Tichy

Abstract

In this paper the Quasi-Monte Carlo methods known for Runge Kutta solution techniques of ordinary differential equations and recently also of delay differential equations of one retarded argument are extended to delay differential equations of an arbitrary (but finite) number of retarded arguments. Their convergence and its order are proved, and an extensive numerical investigation is carried out.

1 Introduction

When dealing with real-world problems, the change of a process $y(t)$ and thus the derivative $y'(t)$ in its mathematical representation often not only depend on the value of the process at present, but also on the past values. The differential equations describing such processes are usually called delay or retarded differential equation, since they also involve terms of the form $y(t-\tau(t))$, where $\tau(t)$ is a function with positive function values, the simplest being a constant retardation $y(t-\tau)$ with $\tau > 0$. In [7] we looked at a class of delay differential equations with one retarded argument of the form

$$y'(t) = f\left(t, y(t), y(t-\tau(t))\right), \quad \text{for } t \geq t_0, \tag{1}$$

where the solution $y(t)$ is a d-dimensional real-valued function, $\tau(t)$ is the continuous delay function, which was assumed to be bounded from below by $\tau_0 > 0$). Furthermore, $\phi(t)$ is the initial function, which is piecewise continuous at least on the interval $(\inf_{t_0 \leq t}(t-\tau(t)), t_0)$. The analysis there was restricted to the case when $\tau(t)$ fulfills the condition $t_1 - \tau(t_1) \leq t_2 - \tau(t_2)$ for $t_1 \leq t_2$.

In that paper [7] we generalized the known Runge-Kutta (Quasi-) Monte Carlo methods, henceforth called RK(Q)MC methods in short, as proposed by Stengle [17, 18],

*This research was supported by the Austrian Science Fund Project S-8308-MAT

2000 Mathematics Subject Classification. Primary 65C05; Secondary 65L06

Key words and phrases. Delay differential equation, Quasi-Monte Carlo method, Runge Kutta scheme, Hermite interpolation, Runge-Kutta Quasi-Monte Carlo

Lécot [9], Coulibaly and Lécot [1] and Lécot and Koudiraty [10] for ordinary differential equations, to delay differential equations of the form (1). Such randomized Runge Kutta algorithms are obtained similar to conventional Runge Kutta methods, except that for conventional Runge Kutta methods $f(t, y(t))$ is Taylor-expanded in both t and $y(t)$, while for (Quasi-) Monte Carlo Runge Kutta schemes the expansion is done only in $y(t)$, resulting in an integral equation. This equation is then solved using conventional (Quasi-) Monte Carlo integration. The advantage of these schemes is that they no longer put any smoothness requirements in t on the function $f(t, y(t))$ or $f(t, y(t), y(t-\tau(t)))$, but instead f needs to be only of bounded variation in time t. However, the solution needs to be piecewise $r/2$-times differentiable to be able to use Hermite interpolation. Although the RKQMC methods are more expensive than the conventional Runge-Kutta methods due to the use of (Quasi-) Monte Carlo integration, for heavily oscillating differential equations the RKQMC methods lead to a significantly reduced error compared to Runge Kutta schemes. As we will show in the last section, RKQMC methods can even be applied to delay differential equations, where classical solution methods become unstable. Furthermore, the integration points for the (Q)MC integration can be calculated in parallel, and thus make use of the power of parallel computers, which is not easily possible for conventional Runge Kutta schemes. As we showed in [7], the low order RKQMC methods can even outperform high order Runge Kutta schemes, if the delay differential equation (DDE) varies significantly faster in t than it does in $y(t)$.

2 Description of the problem

In this paper we will consider Quasi-Monte Carlo Runge Kutta methods for a generalization of the delay differential equation (1) to an arbitrary (but finite) number of retarded arguments:

$$\begin{aligned} y'(t) &= f\left(t, y(t), y(t-\tau_1(t)), \ldots, y(t-\tau_k(t))\right), \quad \text{for } t \geq t_0,\, k \geq 1, \\ y(t) &= \phi(t), \text{ for } t \leq t_0\ , \end{aligned} \tag{2}$$

where the solution $y(t)$ is a d-dimensional real-valued function, $\tau_1(t), \ldots, \tau_k(t)$ are continuous delay functions, which we assume to be bounded from below by $\tau_0 > 0$. Furthermore, $\phi(t)$ is the initial function, which shall be continuous at least on the interval $[\inf_{t_0 \leq t}(t-\tau(t)), t_0]$.

Like in [7], we will sequentially calculate the approximated function values y_n at times $t_n = t_0 + \sum_{j=0}^{n} h_j$ for time steps h_n. To obtain the values of the retarded arguments $y(t-\tau_j(t))$ from the sequence (y_n) we will use Hermite interpolation, because the numerical value of the derivative $y'(t_j)$ at time t_j is known from the differential equation. The main advantage of Hermite interpolation over interpolation methods using just the function values is that it only needs half the points to obtain a given interpolation order.

For f continuous in t and Lipschitz in the other variables, Driver [2] gave a local existence and uniqueness theorem for a very general Volterra functional delay differential equation, which contains (2) as a special case. In this case, the existence theorem reads:

Lemma 1 (local existence and uniqueness, Driver [2]). *Let f be (i) continous in t and (ii) locally Lipschitz in the other arguments $y(t)$ and $y(t-\tau_k(t))$, and the initial function $\phi(t)$ continuous on $[\alpha, t_0]$, where $\alpha = \inf_{\substack{t\geq t_0 \\ 1\leq j\leq k}} (t-\tau_j(t))$. Then for sufficiently small $h > 0$ there exists a unique solution $y(t; t_0, \phi)$ to the differential equation (2) for $\alpha \leq t < t_0 + h$.*

Remark. One should notice that while the RKQMC methods might still be applicable, if f is not continous but only bounded and measurable in t, this theorem cannot be applied. However, under hypotheses similar to the ones Stengle assumes in [18, Hypothesis 2.1], Picard iteration can be applied to assure local existence and uniqueness if f is only bounded and measurable in t.
If f is not globally continuous, however, Hermite interpolation and thus RKQMC methods for DDE can only be applied, if the solution y is at least piecewise continuous on intervals which contain $p/2$ or more support points $t_j, \ldots, t_{j+p/2}$ so that the interpolation error can be bounded using the interpolation order.

However, even smoothness of f and ϕ does not guarantee smoothness of the solution $y(t)$, because similar to delay differential equations with one retarded argument (see [13]) the solution in general will have discontinuous first derivatives at times $t_j^{(1)}$ which are defined as the solutions of the equations $t_j^{(1)} - \tau_j(t_j^{(1)}) = 0$ for $1 \leq j \leq k$, discontinuous second derivatives at times $t_{j,l}^{(2)}$ with $t_{j,l}^{(2)} - \tau_l(t_{j,l}^{(2)}) = t_j^{(1)}$ for $1 \leq j, l \leq k$, and so on. On the intervals between them, a calculation of the exact solution is possible in theory by inserting the already calculated solution from the previous intervals to get rid of the retarded arguments. Thus the calculation of an exact solution means sequentially solving ordinary differential equations on each of the intervals, which quickly becomes an analytically intractable problem.

In this paper, however, we will not be concerned with exact solutions, but with its numerical approximation by means of (quasi-)randomized Runge Kutta methods. We will first present our RKQMC method, which combines the RKQMC methods by Stengle, Lécot, Koudiraty and Coulibaly with the Hermite interpolation method for delay differential equations. We will give a short convergence proof for arbitrary RKQMC schemes applied to delay differential equations of multiple retarded arguments under certain technical conditions on the delay differential equation, the interpolation and the specific RKQMC scheme taken from ordinary differential equations. A convergence proof for delay differential equations with only one retarded argument was already given by the authors in [7]. The biggest part of the paper, however, will be dedicated to an extensive numerical investigation of the RKQMC methods for delay differential equations.

2.1 Quasi-Monte Carlo methods and low-discrepancy sequences

Quasi-Monte Carlo integration methods have been developed for several reasons. They were developed similar to Monte Carlo methods, only that instead of real (pseudo-) random numbers one uses deterministic sequences, also called low-discrepancy sequences, which possess very good uniform distribution properties instead of good randomness. The use of such sequences allows the calculation of deterministic upper error bounds instead of only probabilistic bounds when using Monte Carlo methods. Furthermore, it turned out, that by using these well distributed sequences, one can significantly increase the order of the integration error to $\mathcal{O}\left(\frac{(\log N)^s}{N}\right)$ compared to an order of $\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$ for Monte Carlo methods. And still QMC methods have all the properties for which Monte Carlo integration is chosen over conventional numerical integration methods. In particular, the accuracy of the integration can be improved at any time by simple adding additional summands without the need to recalculate the rest.

The quality of the uniformity of such an s-dimensional sequence S with N element is usually measured by means of the so-called discrepancy, which is defined as

$$D_N(S) = \sup_{a,b\in[0,1]^s} \left| \frac{A([a,b),S)}{N} - \lambda_s([a,b)) \right|, \tag{3}$$

where $A(E,S)$ counts the number of points of the N-element set S that lie inside the interval E, and $\lambda_s(E)$ denotes the s-dimensional Lebesgue measure. Instead of all half-open intervals of the unit cube, the supremum is often taken over all intervals of the form $[0,a)$ with $a \in [0,1]$, and the associated discrepancy is called the star discrepancy $D_N^*(S)$.

The notion of discrepancy is especially important because the integration error of functions f of bounded variation $V(f)$ on $[0,1]^s$ in the sense of Hardy and Krause (see e.g. [3]) can be bounded by the famous Koksma-Hlawka inequality:

$$\left| \frac{1}{N}\sum_{n=1}^{N} f(\mathbf{x}_n) - \int_{[0,1]^s} f(\mathbf{u})d\mathbf{u} \right| \leq V(f)D_N^*(\mathbf{x}_1,\ldots,\mathbf{x}_N) \tag{4}$$

for any point set $S = \{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$. For a proof of this famous error bound of Quasi-Monte Carlo integration, we refer to [3]. In contrast to Monte Carlo error bounds, this inequality is a deterministic error bound which holds for all sets S.

Especially for dimensions s not too large, the use of so-called low-discrepancy sequences, i.e. sequences with discrepancy of order

$$D_N^*(\mathbf{x}_1,\ldots,\mathbf{x}_N) \leq C_s \frac{(\log N)^s}{N} \tag{5}$$

with in explicitly computable constant C_s, can reduce the integration error considerably, even compared to conventional Monte Carlo methods. This discrepancy order with respect to N is conjectured to be optimal, however, known values for C_s are usually usually too pessimistic, and the dependency on s varies heavily for the known bounds (for a profound and very general investigation on the dependency on s, especially for large s, see [12]).

Examples of such low-discrepancy sequences are Halton sequences and net sequences, with Sobol's, Faure's and Niederreiter's sequences being the most prominent examples.

The n-th element of the s-dimensional *Halton Sequence* [6] in pairwise prime integer bases $b_1, \ldots, b_s$ is defined as $\xi_n = (b_{p_1}(n), \ldots, b_{p_s}(n))$, where $b_p(n)$ is the digit reversal function of the representation of n in base p, i.e.

$$b_p(n) = \sum_{k=0}^{\infty} n_k p^{-k-1} \quad \text{where } n = \sum_{k=0}^{\infty} n_k p^k \ ,$$

which means that the p-adic expansion of n is reversed at the comma. The proofs by Halton [6] showed that the discrepancy is minimal if the p_i are chosen as the s smallest prime numbers.

Even better theoretic error bounds can be obtained by (t, s)-sequences in a chosen base p, which are based on (t, m, s)-nets in base p. An s-dimensional point set $\mathcal{P}$ with b^m elements is called a (t, m, s)-*net* if every elementary interval J (which is an interval of the form $J = \prod_{i=1}^{s} \left[a_i p^{-d_i}, (a_i + 1)p^{-d_i}\right)$ with $a_i, d_i \in \mathbb{N}$) of Lebesque measure $\lambda_s(J) = p^{t-m}$ contains exactly p^t elements of the set.

Using this definition of a (t, m, s)-net, a sequence $S = \{\xi_1, \xi_2, \ldots\}$ of points in $[0, 1]^s$ is called a (t, s)-*sequence*, if for all integers $k \geq 0$ and $m > t$ the point sets consisting of the ξ_n with $kp^m < n \leq (k + 1)b^m$ forms a (t, m, s)-net in base b.

The discrepancy of the (t, s)-sequence is minimal for t being as small as possible, but unfortunately, $(0, s)$-nets do not exist for all bases b. See for example [11] for lower bounds of t for given pairs (s, b). Examples of (t, s) nets are:

- *Sobol sequences* are (t, s)-sequences in base 2 with values for t depending on s. They were initially proposed by Sobol [16], but other choices for the direction numbers used in the construction have also been proposed (e.g. [15]).
- *Faure sequences* [4] are $(0, s)$-sequences in a base b which is the smallest prime number fulfilling $b \geq s$. The s-dimensional Faure sequence is defined by

  $$\left\{\phi_b(n), P(\phi_b(n)), \ldots, P^{s-1}(\phi_b(n))\right\}$$

  where the function P is defined for a b-adic rational $x \in [0, 1]$ with expansion $x = \sum_{j=0}^{\infty} x_j b^{-j-1}$ as $P(x) = \sum_{j=0}^{\infty} \xi(x_j) b^{-j-1}$ with $\xi(x_j) = \sum_{i \geq j} \binom{i}{j} x_i \mod b$.

- *Niederreiter sequences* [11] are a generalization of Sobol's and Faure's sequences and yield general (t, s)-sequences for arbitrary bases p with the restrictions on t mentioned above. For various different constructions see [11].

3 The RKQMC method for differential equations with multiple retarded arguments

Following the path of Stengle [18] as well as Lécot, Coulibaly and Koudiraty ([9, 1, 10], we will now deduce Quasi-Monte Carlo schemes for delay differential equations with multiple retarded arguments. These schemes are akin to the family of Runge-Kutta schemes and thus they are often called Runge Kutta (Quasi-) Monte Carlo schemes (RKQMC in short).

In [10] Lécot and Koudiraty treated the case of ordinary differential equations and derived a quasi-randomized Runge Kutta scheme. If we assume that we already know the exact or approximated solution $\hat{y}$ to the equation up to time t (or at least up to $\max_{1\leq i\leq k}(t-\tau_k(t))$), this solution can be inserted into the right hand side of equation (2), and the equation simplifies to an ordinary differential equation at time t with $g(t, y(t)) := f(t, y(t), y(t-\tau_1(t)), \dots, y(t-\tau_k(t))$. The value of the solution at this time can now simply be calculated by such a Runge Kutta QMC scheme. For this reason, in the following paragraphs we will repeat Koudiraty's arguments which lead to RKQMC methods for ordinary differential equations. Throughout the whole argumentation, the existence and uniqueness of the solution $y(t)$ with $y' \in L^1(0, T)$ needs to be assumed, which in our case is guaranteed by the theorem and the remarks of the previous section.

Starting from the differential equation $y'(t) = g(t, y(t), y(t-\tau_1(t)), \dots)$ or its equivalent integral representation

$$y(t_0 + h) = y(t_0) + \int_{t_0}^{t_0+h} g(u, y(u), y(t-\tau_1(t)), \dots)du$$

the retarded values $y(t-\tau_1(t)), \dots, y(t-\tau_n(t))$ are approximated by ordinary Hermite interpolation from the already calculated values at the discrete times $t_n \leq t$. This is done using an interpolation function

$$\tilde{z}(t) = z_{(y_i)_{i\leq n}}(t) = \begin{cases} \phi(t), & \text{if } t \leq t_0 \\ P_q\left(t; (y_i), (y_i')\right) & \text{otherwise} \end{cases} \tag{6}$$

which uses the initial function $\phi(t)$ for all values prior to the starting value. For later times the function $z(t)$ is constructed piecewise by Hermite-interpolation where appropriate grid points t_n are used as support points and the value of $y'(t_n)$ is approximated

by $f(t_n, y(t_n), y(t_n-\tau_1(t_n), \ldots, y(t_n-\tau_n(t_n)))$. The support points to construct the Hermite polynomial for a certain value of t are chosen such that extrapolation is avoided, and all support points lie inside the same interval of smoothness as the time t. Naturally, it is of advantage to use adjacent grid points with the value of t in between, so that the interpolation error is kept to a minimum. Apart from this incentive, the method does not put any further restrictions on the choice, except that the resulting function $g\,(t, y(t); z(t-\tau_1(t)), \ldots, z(t-\tau_n(t)))$ needs to be of bounded variation in t to be able to apply the RKQMC methods at all. In contrast to previous works (e.g. Oberle and Pesch [13]), smoothness in the interpolation function is thus not required, which would mean that only one interpolation polynomial has to be used for all interpolation values in one time step. Since our method involves integration over a whole interval, this would involve an extensive amount of extrapolation, so the possibility to chose the interpolation polynomial freely according to only the value of t is of vital importance here.

Inserting the interpolation function $z(t)$ into the delay differential equation transforms it into an ordinary differential equation, or an ordinary integral equation, respectively:

$$y(t_0+h) = y(t_0) + \int_{t_0}^{t_0+h} g(u, y(u), z(u-\tau_1(u)), \ldots)du \tag{7}$$

Starting from equation (7), the function g is Taylor-expanded up to order s with respect to $y(t)$ and recursively inserted into itself. Since this requires g to be s-times differentiable in y, the function f needs to be s-times differentiable in y. According to the discussion above for smooth f and ϕ the solution $y(t)$ is also smooth on the interval $[t_0, \min_{i\le k}\tau_i)$. Note that g does not have to be Taylor-expanded in the retarded arguments $y(t-\tau_1(t), y(t-\tau_2(t)), \ldots$, because their values are already calculated via Hermite interpolation.

Using the general identity

$$\left(\int_{t_0}^{u_i} f(u)du\right)^n = n! \int_{t_0}^{u_i} \cdots \int_{t_0}^{u_{i+n}} f(u_{i+n-1}) \cdot \cdots \cdot f(u_{i+n})du_{i+n}du_{i+1}$$

this gives [8]

$$\begin{aligned} y(t_0+h) = y(t_0) &+ \int_{t_0}^{t_0+h} F_1(u_1; y)du_1 + \int_{t_0}^{t_0+h}\int_{t_0}^{u_1} F_2(u_1, u_2; y)du_2du_1 + \ldots \\ &+ \int_{t_0}^{t_0+h}\int_{t_0}^{u_1}\int_{t_0}^{u_{s-1}} F_s(u_1, \ldots, u_s; y)du_s \ldots du_1 \\ &=: \frac{1}{s!h^{s-1}} \int_{t_0}^{t_0+h}\int_{t_0}^{t_0+h} G_s(\bar{u}_1, \ldots \bar{u}_s; y)du_s \ldots du_1 \end{aligned}$$

with

$$F_i(u_1, \ldots, u_i; y) := D_y^1 F_{i-1}(u_1, \ldots, u_{i-1}; y) f(u_i; y(u_i), y(u_i-\tau_1(u_i)), \ldots)$$

and $F_0(y) := y$. Here $\bar{u}$ denotes the vector u with the components sorted in ascending order. The function G_s still contains derivatives of $f(t; y, y(t-\tau_1(t)), ...)$ with respect to $y(t)$. Using a specific identity like the ones proposed by Stengle [18] or Lécot and Koudiraty [8] to approximate linear combinations of derivatives of f by Runge-Kutta-like schemes of order s, one can obtain particular increment functions

$$\tilde{G}_s(\bar{u}_1, \dots \bar{u}_s; y(t), y(t-\tau_1(t)), \dots, y(t-\tau_n(t)))$$

which approximate the exact increment function to order h^{s+1}:

$$\tilde{G}_s = G_s(\bar{u}_1, \dots \bar{u}_s; y(t), y(t-\tau_1(t)), \dots, y(t-\tau_n(t))) + \mathcal{O}(h^{s+1}) \ .$$

As a final step to obtain the method, the remaining integral over $[t_0, t_0+h)^s$ is calculated by Quasi-Monte Carlo integration

$$\int_{t_0}^{t_0+h} \int_{t_0}^{t_0+h} G_s(\bar{u}_1, \dots \bar{u}_s; y) \approx \frac{1}{N} \sum_{i=1}^{N} G_s\left(t^{(\bar{n})}; y\right) \tag{8}$$

Stengle [17, 18] proposed a scheme of second order which uses pseudo-random numbers, while the schemes proposed by Lécot (first and second order, [9]), Coulibaly and Lécot (second order, [1]) and Lécot and Koudiraty (third order, [8]) use quasi-random numbers and thus absolute upper error bound were given for these schemes.

In the calculations presented this paper, we will use these existing first, second and third order RKQMC schemes for the increment function $G_s(t, y(t), y(t-\tau_1(t)), \dots, y(t-\tau_n(t))$ and not derive our own schemes.

4 Convergence of the method

Before we can prove convergence of the method, we have to define it in a more rigorous manner. For brevity, we will suppress the argument t of the function y whenever it is clear what the argument is.

We first rewrite the delay differential equation (2) in a more Volterra functional equation-like form

$$\begin{aligned} y'(t) &= f(t, y(t), z(t)) && t \geq t_0 \\ z(t) &= (Fy)(t) \\ y(s) &= g(s) && s \leq t_0 \end{aligned} \tag{9}$$

where all dependence on retarded values is moved into $(Fy)(t)$. We will furthermore assume that both F and f are Lipschitz continuous in all arguments except t.

Our method, which will generate a sequence (y_n) of values that approximate the exact solution $y(t)$ at the grid points $t_0 \le t_1 \le t_2 \le \cdots \le t_m$, then reads

$$\begin{aligned} y_{n+1} &= y_n + h_n \sum_{i=1}^{N} G_s\left(t_{n,i}; y_n, \tilde{z}(t)\right) \\ \tilde{z}(t) &= (\tilde{F} y_j)(t) \end{aligned} \tag{10}$$

for $n \ge 0$, and $y_j = \phi(t_j)$ for $j < 0$. The function $(\tilde{F} y_j)$ uses the interpolation function (6) for the retarded values of y_j.

Our goal is now to establish uper bounds for the error $\|e_{j+1}\| = \|y_{j+1} - y(t_{j+1})\|$. We will henceforth only use RKQMC methods which converge with order $\mathcal{O}(h^p)$ for ordinary differential equations. Following the idea of Oppelstrup [14] for the solution of DDE using Hermite interpolation, the following theorem proves convergence of our method for DDE:

Theorem 1. *Let $y(t)$ be the solution of the DDE (9), and $(y_j)_{0\le j\le n}$ the approximate solution obtained by (10) on the grid $t_0 \le t_1 \le \cdots \le t_n$. Let furthermore $\left\|E_G^{ODE}\right\|_j$ be the error in the j-th step of the RKQMC method applied to the ODE which uses the exact solution for the retarded values, and $\left\|E_G^{ODE}\right\| := \max_j \left\|E_G^{ODE}\right\|_j$. The conventional Hermite interpolation error $\left\|E_r^{interpol}\right\|_j$ is defined as $\max_{t_j \le u \le t_{j+1}} (Fy(t))(u) - (\tilde{F}y(t))(u)$, and $\left\|E_r^{interpol}\right\| := \max_{0\le j\le n} \left\|E_r^{interpol}\right\|_j$. If*

1. *the RKQMC method with increment function $G_s(t_n, y_n)$ converges with order p for ODE as $h_j \to 0$,*
2. *G_s as defined in (10) is Lipschitz in $\tilde{z}$ with constant $\tilde{\mathcal{L}} > 0$, and F is Lipschitz with constant $\mathcal{L}_2 > 0$,*
3. *$\tilde{F}$ fulfills a Lipschitz criterion $\left\|(\tilde{F}u_j)(t) - (\tilde{F}v_j)(t)\right\| \le \tilde{\mathcal{L}} \max \|u_k - v_k\|$ with Lipschitz constant $\tilde{\mathcal{L}} > 0$,*
4. *the Hermite interpolation has order r,*
5. *and the initial error $\|e_0\|$ vanishes,*

then the method converges, and the error is bounded by

$$\|e_{j+1}\| \le \|e_0\| e^{\mathcal{L}t_j} + \frac{\left(e^{t_j\mathcal{L}} - 1\right)}{\mathcal{L}} \left(\left\|E_G^{ODE}\right\| + \mathcal{L}_1\mathcal{L}_2 \left\|E_r^{interpol}\right\|\right)$$

Remark. Like in the case of one retarded argument [7], this theorem shows that the RKQMC1, RKQMC2 and RKQMC3 methods by Lécot, Coulibaly and Koudiraty ([9, 1, 8]) can be successfully applied to DDE. We refrain from giving explicit upper bounds for the error, since these are usually very generous and several orders of magnitude (up to 10 orders) above the numerically experienced errors. For this reason, error bounds as obtained above are useful to prove convergence, but not to give a good error estimate.

Remark. Condition 4 puts an effective upper bound on the step size, namely that $\frac{r}{2}$ Points $t_k, ..., t_{k+\frac{r}{2}}$ need to be inside an interval of smoothness as discussed in chapter 2.

Proof of theorem 1. Using the definition of our method, we get for the error $\|e_{j+1}\|$:

$$\|e_{j+1}\| = \left\| e_j + h_h \left(\hat{G}(t_j, y(\cdot)) - \sum_{i=1}^{N} G_s(t_{j,i}; y_j, \tilde{z}(t_{j,i})) \right) \right\| ,$$

where $\hat{G}(t_j, y(\cdot)) = \frac{1}{h_j} \int_{t_j}^{t_j+h_j} g\left(u, y(u), y(u-\tau_1(u)), \dots y(u-\tau_n(u))\right) du$ is the exact increment function using the solution $y(\cdot)$ for the retarded values. This leads to the further estimate

$$\begin{aligned}
\|e_{j+1}\| &\leq \|e_j\| + h_j \left\| \hat{G}(t_j, y(\cdot)) - \frac{1}{N} \sum_{i=1}^{N} G_s(t_{j,i}; y(t_j), z(\cdot)) \right\| + \\
&\quad + h_j \left\| \frac{1}{N} \sum_{i=1}^{N} G_s(t_{j,i}; y(t_j), z(\cdot)) - \frac{1}{N} \sum_{i=1}^{N} G_s(t_{j,i}; y_j, z(\cdot)) \right\| + \\
&\quad + h_j \left\| \frac{1}{N} \sum_{i=1}^{N} G_s(t_{j,i}; y_j, z(\cdot)) - \frac{1}{N} \sum_{i=1}^{N} G_s(t_{j,i}; y_j, \tilde{z}(\cdot)) \right\| \\
&\leq \|e_j\| + h_j \left\| E_G^{\mathrm{ODE}} \right\|_j + h_j \frac{1}{N} \sum_{i=1}^{N} \|G_s(t_{j,i}, y_j, y(\cdot)) - G_s(t_{j,i}, y_j, \tilde{z}(\cdot))\|
\end{aligned}$$

where $\left\| E_G^{\mathrm{ODE}} \right\|_j$ denotes the error in the j-th step of the RKQMC method applied to the ODE using the exact solution for the retarded arguments. The error due to the interpolation can furthermore be estimated by

$$\begin{aligned}
&\|G_s(t_{j,i}, y_j, z(\cdot)) - G_s(t_{j,i}, y_j, \tilde{z}(t_j))\| \leq \\
&\leq \left\| G_s(t_{j,i}; y_j, z(\cdot)) - G_s(t_{j,i}; y_j, \tilde{z}_{(y(t_i))_{i\leq j}}(\cdot)) \right\| + \left\| G_s(t_{j,i}; y_j, \tilde{z}_{(y(t_i))_{i\leq j}}(\cdot)) - G_s(t_{j,i}, y_j, \tilde{z}(\cdot)) \right\| \\
&\leq \mathcal{L}_1 \mathcal{L}_2 \left(\left\| y(\cdot) - \tilde{z}_{(y(t_i))_{i\leq j}}(\cdot) \right\| + \left\| \tilde{z}_{(y(t_i))_{i\leq j}}(\cdot) - \tilde{z}(\cdot) \right\| \right) \\
&\leq \mathcal{L}_1 \mathcal{L}_2 \left(\left\| E_r^{\mathrm{interpol}} \right\|_j + \tilde{\mathcal{L}} \max_{0\leq k\leq j} \|e_k\| \right)
\end{aligned}$$

using the Lipschitz conditions on the third argument of G_s with constant $\mathcal{L}_1$ and on F with constant $\mathcal{L}_2$.

Since the $\|e_i\|$ are monotone increasing, $\max_{0\le k\le j}\|e_k\| = \|e_j\|$. If we now put everything together, we get

$$\begin{aligned}
\|e_{j+1}\| &\le \|e_j\| + h_j \left\|E_G^{\mathrm{ODE}}\right\|_j + h_j\mathcal{L}_1\mathcal{L}_2\left(\left\|E_r^{\mathrm{interpol}}\right\|_j + \tilde{\mathcal{L}}\,\|e_j\|\right)\\
&\le \|e_j\|\left(1+\mathcal{L}_1\mathcal{L}_2\tilde{\mathcal{L}}h_j\right) + h_j\left(\left\|E_G^{\mathrm{ODE}}\right\| + \mathcal{L}_1\mathcal{L}_2\left\|E_r^{\mathrm{interpol}}\right\|\right)\\
&=: \|e_j\|\,(1+\mathcal{L}h_j) + h_j\mathcal{C}\\
&\le \|e_0\|\prod_{k=0}^{j}(1+h_k\mathcal{L}) + \mathcal{C}\sum_{n=0}^{j}h_n\prod_{k=n+1}^{j}(1+h_k\mathcal{L}) \qquad (11)\\
&\le \|e_0\|\,e^{\mathcal{L}\sum_{k=0}^{j}h_k} + \mathcal{C}\sum_{n=0}^{j}\left(\frac{1+h_n\mathcal{L}}{\mathcal{L}}\prod_{k=n+1}^{j}(1+h_n\mathcal{L}) - \frac{1}{\mathcal{L}}\prod_{k=n+1}^{j}(1+h_n\mathcal{L})\right) =\\
&= \|e_0\|\,e^{\mathcal{L}t_j} + \frac{\mathcal{C}}{\mathcal{L}}\left(\prod_{k=0}^{j}(1+h_n\mathcal{L}) - 1\right) \le \|e_0\|\,e^{\mathcal{L}t_j} + \frac{\mathcal{C}}{\mathcal{L}}\left(e^{t_j\mathcal{L}}-1\right)\\
&= \|e_0\|\,e^{\mathcal{L}_1\mathcal{L}_2\tilde{\mathcal{L}}t_j} + \frac{\left(e^{t_j\mathcal{L}_1\mathcal{L}_2\tilde{\mathcal{L}}}-1\right)}{\mathcal{L}_1\mathcal{L}_2\tilde{\mathcal{L}}}\left(\left\|E_G^{\mathrm{ODE}}\right\| + \mathcal{L}_1\mathcal{L}_2\left\|E_r^{\mathrm{interpol}}\right\|\right) \qquad (12)
\end{aligned}$$

where in (11) the inequality was recursively inserted into itself, and an empty product should be understood as 1.

From the last inequality (12) the convergence of the method is obvious, and it follows that for a vanishing initial error $\|e_0\|$ the convergence of the RKQMC method for DDE is $\min\{p, r\}$, where p is the convergence order of the RKQMC method for ODE, and r is the order of the interpolation error. ■

5 Numerical Experiments

The idea of using Quasi-Monte Carlo integration for the integration over t stems from the attempt to minimize the error for heavily oscillating differential equations (or delay differential equations with heavily oscillating solutions, which results in a heavily oscillating ODE after inserting the solution $y(t)$ for the retarded values). Instead of accepting the (possibly large) error at just one retarded value, the dependence on t is averaged out, and the error thus minimized. Since RKQMC methods do not exhibit any advantage over conventional methods for non-oscillating equations, we will only investigate several heavily oscillating differential equations.

In our examples we will compare the first, second and third order RKQMC methods with some conventional Runge Kutta schemes with Hermite interpolation as proposed for example by Oppelstrup [14] or Oberle and Pesch [13]. In particular, as low-order

$$
\begin{array}{c|ccc}
0 & & & \\
\frac{1}{3} & \frac{1}{3} & & \\
\frac{2}{3} & 0 & \frac{2}{3} & \\
\hline
 & \frac{1}{4} & 0 & \frac{3}{4}
\end{array}
\qquad
\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & \frac{1}{2} & & & \\
1 & 0 & 1 & & \\
1 & 0 & 0 & 1 & \\
\hline
 & \frac{1}{6} & \frac{2}{3} & 0 & \frac{1}{6}
\end{array}
$$

$$
\begin{array}{c|ccccccc}
0 & & & & & & & \\
\frac{1}{2} & \frac{1}{2} & & & & & & \\
\frac{2}{3} & \frac{2}{9} & \frac{4}{9} & & & & & \\
\frac{1}{3} & \frac{7}{36} & \frac{2}{9} & -\frac{1}{12} & & & & \\
\frac{5}{6} & -\frac{35}{144} & -\frac{55}{36} & \frac{35}{48} & \frac{15}{8} & & & \\
\frac{1}{6} & -\frac{1}{360} & -\frac{11}{36} & -\frac{1}{8} & \frac{1}{2} & \frac{1}{10} & & \\
1 & -\frac{41}{260} & \frac{22}{13} & \frac{43}{156} & -\frac{118}{39} & \frac{32}{195} & \frac{80}{39} & \\
\hline
 & \frac{13}{200} & 0 & \frac{11}{40} & \frac{11}{40} & \frac{4}{25} & \frac{4}{25} & \frac{13}{200}
\end{array}
$$

Table 1: Butcher tableaus (taken from [5]) for Heun's (third order, 3-step) and Runge's (third order, 4-step) classical Runge-Kutta schemes, as well as for Butcher's high-order scheme (6-th order, 7 step)

Runge-Kutta schemes we applied the well-known 4-stage Runge scheme of order 3 and the 3-stage method of Heun of order 3, while as a high-order Runge-Kutta scheme we use Butcher's 6-th order, 7-stage method as described in [5]. The Butcher tableaus of these classical Runge Kutta schemes are given in Figure 1 for reference.

As a measure of quality of the obtained numerical solutions we will use the sum of deviations from the exact solution at the times $t = 0 \ldots 0.1 \ldots 20$, i.e.

$$S^{\lambda,(\mathrm{meth})} := \sum_{i=1}^{200} \left| y\left(\frac{i}{10}\right) - y^{(\mathrm{meth}),\lambda}_{\frac{i}{10}} \right| , \tag{13}$$

and all graphical comparisons on the dependence on λ will be given in terms of least-squares fits of the functions $\{1, x, x^2, \log(x)\}$ to the Functions $\log\left(S^{(\mathrm{meth}),\lambda}\right)$. As we mentioned above, the RKQMC methods use a sum of N function values of f in every time step to approximate the integral in (8), so a lot more calculation time is needed for a single RKQMC step than for a conventional Runge-Kutta step. To include this effect into our evaluation, we will also compare the methods using the least-squares fits to the logarithm of the timed error S_T, which we define as

$$S_T^{\lambda,(\mathrm{meth})} := T^{\lambda,(\mathrm{meth})} S^{\lambda,(\mathrm{meth})} , \tag{14}$$

with $T^{\lambda,(\mathrm{meth})}$ being the calculation time for the given method and value of λ, so that two methods, where one needs only half the time, but twice the error of the other, are treated alike.

Unless we mention it explicitely, all values are obtained with a constant time step of $h_n = h = 0.001$ for classical Runge Kutta schemes, and a step of $h = 0.01$ for

RKQMC schemes. The simulation is done up to time $T = 20$, and a forth order Hermite interpolation is used for the retarded values $y(t - \tau_k(t))$. As low-discrepancy sequence for the numerical integration we use Sobol's sequence, where the N numbers for each time step are taken sequentially, i.e. we use the first Nn elements of the sequence to obtain the solution value at time t_n.

As a first example of a delay differential equation with two retarded arguments, we applied our RKQMC methods to the delay differential equation

$$\begin{aligned} y'(t) &= 3y(t-1)\sin(\lambda t) + 2y(t-1.5)\cos(\lambda t), \quad t \geq 0 \\ y(t) &= 1, \quad t \leq 0\ , \end{aligned} \tag{15}$$

which is very similar in its structure to the DDE we investigated in [7]. Its "exact" solutions, which we obtained by a run of Butcher's high order method using a step size of $h_n = 0.00001$, are shown in figure 1.

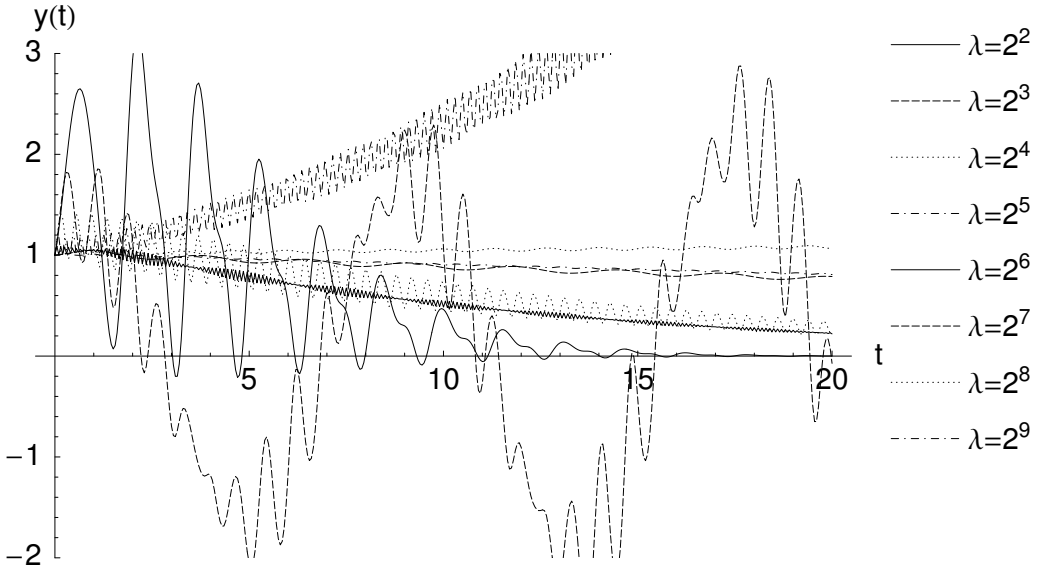

Figure 1: Exact solution of examp(15) for some values of λ obtained by a run with very small step size.

Figure 2 and table 2 show the results of our calculations for increasing $\lambda = 2^k$, $0 \leq k \leq 20$. As one might expect from the the results in [7], for small values of λ, the conventional Runge-Kutta schemes clearly give better results than the quasi-randomized methods presented here. However, for values of λ above 2^{11}, the RKQMC methods become competitive, and give better results for $\lambda > 12$. One has to notice that all three RKQMC methods we investigate show roughly the same behaviour, although in

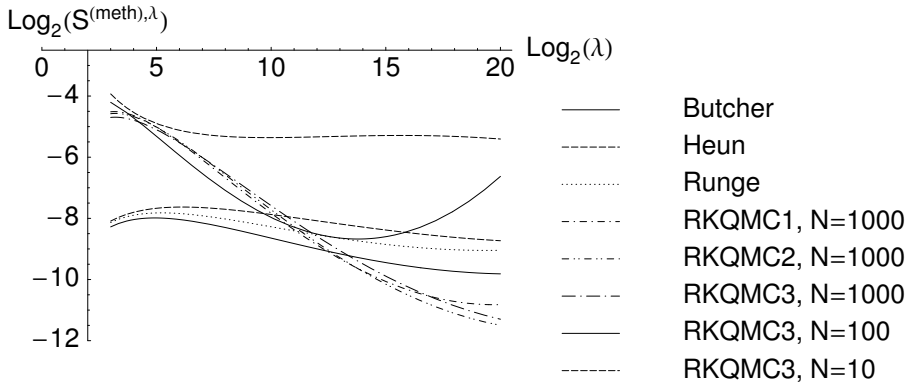

Figure 2: RKQMC vs. conventional Runge-Kutta schemes for equation (15).

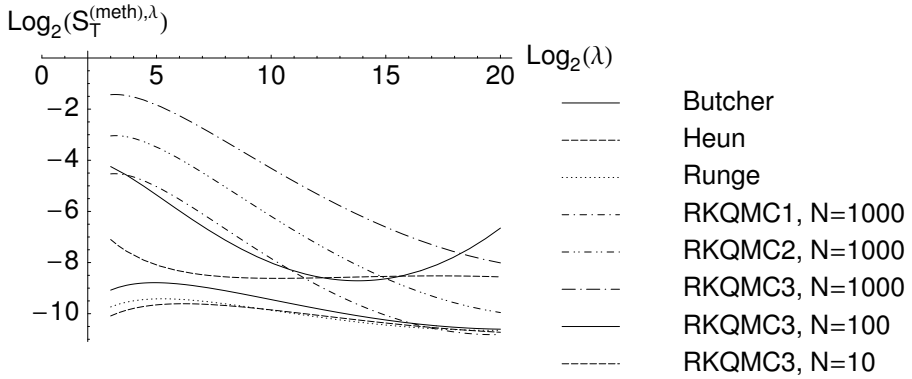

Figure 3: Time-corrected comparison for equation (15).

λ	Butcher	Runge	RKQMC1 $N=1000$	RKQMC2 $N=1000$	RKQMC3 $N=1000$	RKQMC3 $N=10$
2^5	−7.39198	−7.17466	−4.21003	−4.21009	−4.21026	−5.55528
2^6	−9.12876	−8.91401	−6.63217	−6.63697	−6.63852	−5.05383
2^7	−9.39176	−9.16272	−5.71178	−5.7132	−5.71862	−5.64201
2^8	−8.32904	−8.11228	−6.85581	−6.85166	−6.8277	−4.75116
2^9	−7.28245	−7.05248	−5.54155	−5.53954	−5.53957	−5.63427
2^{10}	−8.25863	−8.00724	−6.65457	−6.61329	−6.76324	−4.56086
2^{11}	−7.65107	−7.45674	−8.69076	−8.28356	−8.43429	−6.29302
2^{12}	−8.31722	−8.08087	−9.05153	−8.95747	−8.85853	−2.83075
2^{13}	−9.70187	−9.80577	−9.45962	−9.49673	−9.4834	−8.92099
2^{14}	−10.5467	−8.91969	−11.3958	−11.477	−11.7516	−5.78229
2^{15}	−9.90117	−9.0376	−12.5316	−11.4426	−10.1873	−5.38535
2^{16}	−10.6722	−8.91812	−9.03928	−10.7756	−10.0035	−2.928
2^{17}	−9.77875	−9.56794	−9.40501	−9.2114	−8.78821	−4.24964
2^{18}	−7.38222	−7.32184	−10.7567	−11.135	−10.9163	−5.74555
2^{19}	−9.57111	−9.9538	−9.1594	−9.72612	−9.10414	−5.56226
2^{20}	−10.7389	−9.03267	−12.0962	−12.7167	−13.2895	−6.09049
time	(0.5745s)	(0.331s)	(0.995s)	(2.8885s)	(9.657s)	(0.1075s)

Table 2: Error for increasing values of λ in equation (15). The last row shows the average time needed for the method.

theory they are of different order. However, the number N of sample points is of great importance to the numerical result, especially, taking N too small leads to a large bias in the results. It turned out that about $N \approx 1000$ evaluations for each time step are enough to achieve a sufficient accuracy of the Quasi-Monte Carlo integration involved, so we only present results with $N = 1000$. The comparison above does not yet include the time needed to obtain the solution, which varies greatly with the method. If we compare the methods using the time-corrected error S_T as defined in (14), figure 3 shows that – although RKQMC methods need a lot more function evaluations – especially the RKQMC1 method still can compete with the conventional Runge Kutta schemes for large values of λ.

As a second example, we chose the delay differential equation

$$\begin{aligned} y'(t) &= -\frac{1}{2}\left(3y\left(t-\frac{1}{2}\right)+4\cos\left(y\left(t-\frac{\pi}{10}\right)\right)-5\right)+3\log(\lambda)^2 t\cos(\lambda t), \quad t\geq 0 \\ y(t) &= 1, \quad t<0\,, \end{aligned} \tag{16}$$

which has an oscillating solution with increasing amplitude. Here again, for a slowly oscillating differential equation conventional Runge Kutta schemes are favorable, while RKQMC methods gain considerably for heavy oscillations, as figures 4 and 5 and table 3 show.

As a final example, we applied the RKQMC methods to the following delay differential equation

$$\begin{aligned} y'(t) &= \pi\frac{\lambda}{2}\left(y\left(t-2-\frac{3}{2\lambda}\right)-y\left(t-2-\frac{1}{2\lambda}\right)\right), \quad t\geq 0 \\ y(t) &= sin(\lambda t\pi), \quad t<0\,, \end{aligned} \tag{17}$$

with the exact solution $y(t) = sin(\lambda t\pi)$, so the problem does not have discontinuities in any derivative, since the initial condition is already the solution of the differential equation. Because the k-th derivative of $y(t)$ can only be bounded by λ^k, for large λ the interpolation error can also only be bounded by a power of λ. One thus has to expect an exploding error, and the solution method will be very unstable. An interesting question in this case is if RKQMC methods – although still obtaining an exploding error – behave at least a little better than conventional Runge Kutta. Or in other words, can the averaging over the whole interval $[t_k, t_{k+1})$ delay the explosion of the error? Figures 6 and 7 show that while for conventional Runge Kutta schemes the error increases rapidly already for $\lambda > 2^{10}$ (or even $\lambda > 2^7$), the RKQMC methods stay more or less stable until $\lambda > 2^{13}$, and even then the error is several orders of magnitude smaller than with conventional methods.

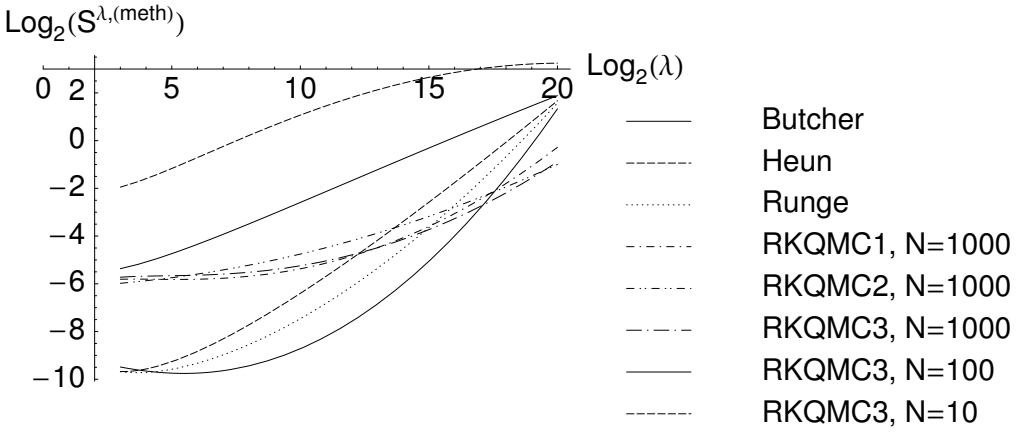

Figure 4: RKQMC vs. conventional Runge-Kutta schemes for equation (16).

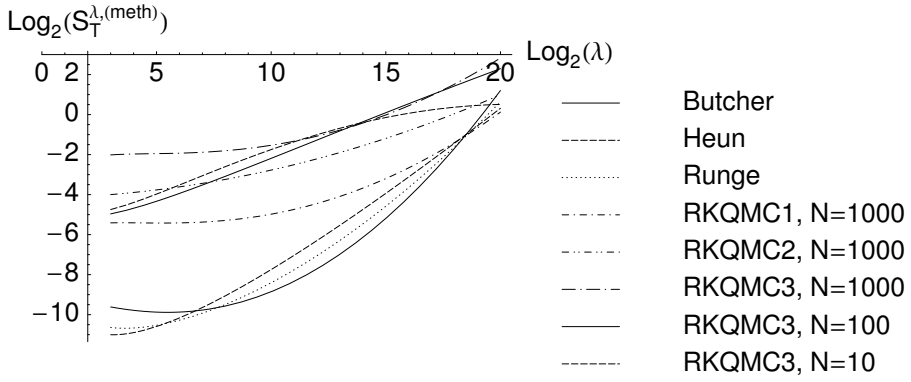

Figure 5: Time-corrected comparison for equation (16).

λ	Butcher	Runge	RKQMC1 $N=1000$	RKQMC2 $N=1000$	RKQMC3 $N=1000$	RKQMC3 $N=10$
2^5	−8.87234	−8.88586	−5.71718	−5.71687	−5.71446	−1.63038
2^6	−9.26313	−9.17274	−5.72643	−5.8531	−5.79093	−1.78973
2^7	−10.1041	−9.86726	−6.31692	−6.26188	−6.02894	−1.98767
2^8	−9.95295	−9.79758	−6.80873	−6.24588	−5.80225	0.31842
2^9	−11.208	−10.7435	−5.94728	−6.03407	−5.95347	0.17748
2^{10}	−11.5016	−11.6134	−5.19338	−3.91158	−4.56006	2.86029
2^{11}	−10.7611	−8.90505	−5.65698	−3.47692	−5.23753	1.29468
2^{12}	−10.6531	−5.24012	−3.95011	−3.41886	−4.53776	3.46736
2^{13}	−5.4083	−1.92602	−6.733	−6.71153	−7.15757	−0.33684
2^{14}	−2.39403	−2.37575	−3.23562	−3.83238	−4.34481	1.77534
2^{15}	−2.04463	−1.2864	−3.86274	−3.65393	−3.62819	2.62794
2^{16}	−2.24314	−1.66459	−1.00946	−0.9048	−1.49998	4.49528
2^{17}	−1.05771	−0.85625	−1.39275	−0.06356	−0.22052	4.10072
2^{18}	−0.42866	−0.42336	−1.11109	−1.2283	−1.71019	3.31797
2^{19}	−1.41526	−1.28994	−0.50625	−0.82309	−1.28871	3.3318
2^{20}	−1.97097	−0.15265	−2.66799	−3.7689	−3.32669	1.8979
time	(0.914s)	(0.52s)	(1.317s)	(3.9515s)	(13.1045s)	(0.144s)

Table 3: Error for increasing values of λ in equation (16)

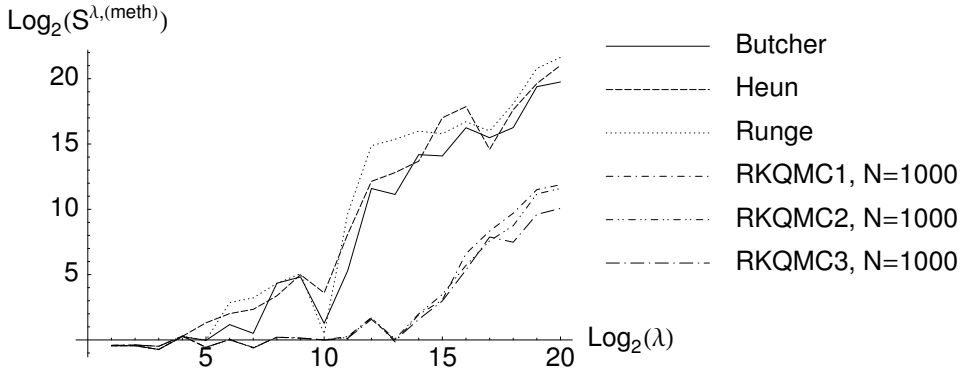

Figure 6: RKQMC can delay numerical unstabilities in heavily oscillating equations like equation (17).

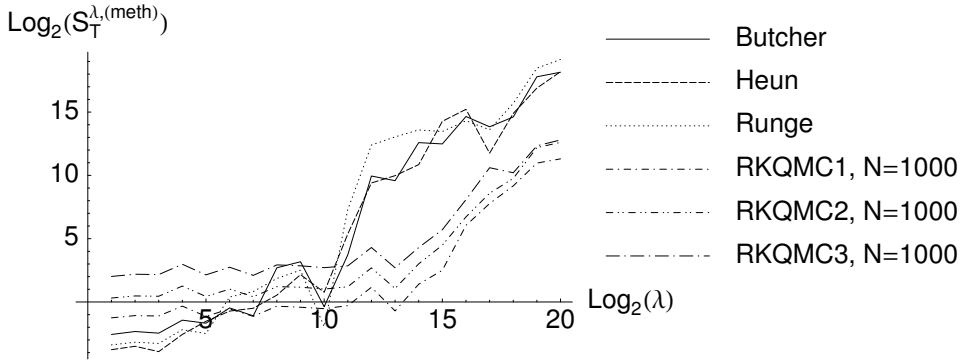

Figure 7: Time-corrected comparison for equation (17).

λ	Butcher	Runge	RKQMC1 $N = 1000$	RKQMC2 $N = 1000$	RKQMC3 $N = 1000$	RKQMC3 $N = 10$
2^5	−0.03254	−0.02311	−0.56156	−0.5624	−0.5612	−0.23613
2^6	1.17245	2.86221	0.02865	0.02788	0.03007	0.33568
2^7	0.50327	3.21856	−0.57879	−0.59252	−0.61976	1.79124
2^8	4.34429	4.33091	0.20416	0.21056	0.20644	1.49253
2^9	4.82238	5.07044	0.13955	0.15332	0.13093	3.3962
2^{10}	1.2717	0.44702	0.00626	0.00308	−0.01275	3.98611
2^{11}	5.32135	9.6919	0.26602	0.19461	0.13048	4.64482
2^{12}	11.5943	14.8719	1.71566	1.68044	1.57747	7.51536
2^{13}	11.1378	15.3605	−0.14943	0.06415	−0.02259	7.20202
2^{14}	14.1873	15.9873	1.93997	1.99741	1.57252	11.7018
2^{15}	14.0843	15.7865	3.05771	3.47916	2.98297	13.5787
2^{16}	16.2482	16.7093	6.6045	5.70631	5.35894	14.4304
2^{17}	15.4825	16.0032	8.33202	7.58254	7.87936	17.3266
2^{18}	16.2581	18.0802	9.70491	8.74674	7.48048	17.3213
2^{19}	19.3875	20.7898	11.5075	11.1776	9.60934	17.2286
time	(0.316s)	(0.1815s)	(0.669s)	(1.977s)	(6.486s)	(0.0725s)

Table 4: All errors for equation (17). RKQMC methods stay stable for $2^8 \leq \lambda \leq 2^{13}$, where conventional Runge Kutta schemes become unstable.

6 Conclusion

In this paper we successfully showed that the RKQMC methods by Stengle, Lécot, Koudiraty and Coulibaly can also be applied to retarded differential equations. While we already showed this in [7] for equations of one delayed argument, here we presented a more general proof and extend it to equations of several retarded arguments. Our numerical investigation showed, that for slowly varying differential equations, conventional Runge Kutta methods have to be preferred over RKQMC schemes, but for heavily oscillating equations or solutions, RKQMC methods can yield better results than conventional schemes. Although the RKQMC schemes are more expensive as far as computing time is concerned, a larger time step can be chosen. Furthermore, RKQMC schemes may stay stable in a region where conventional Runge Kutta schemes already give an exploding error. All in all, RKQMC solution schemes can be viewed as a good complement to classical Runge Kutta schemes with Hermite interpolation for heavily oscillating delay differential equations.

References

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

[2] R. D. Driver. Existence and stability of solutions of a delay-differential system. *Arch. Rational Mech. Anal.*, 10:401–426, 1962.

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

[4] H. Faure. Discrépance de suites associées à un système de numération (en dimension s). *Acta Arithmetica*, 41:337–351, 1982.

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

[6] J. H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multidimensional integrals. *Numer. Math.*, 2:84–90, 1960.

[7] R. Kainhofer. QMC methods for the solution of delay differential equations. submitted, 2002.

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

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

[10] C. Lécot and A. Koudiraty. Numerical Analysis of Runge-Kutta Quasi-Monte Carlo Methods. Submitted, 2001.

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

[12] E. Novak and H. Woźniakowski. When are integration and discrepancy tractable? In *Foundations of computational mathematics (Oxford, 1999)*, pages 211–266. Cambridge Univ. Press, Cambridge, 2001.

[13] H. J. Oberle and H. J. Pesch. Numerical Treatment of Delay Differential Equations by Hermite Interpolation. *Numer. Math.*, 37:235–255, 1981.

[14] J. Oppelstrup. The RKFHB4 Method for Delay-Differential Equations. In R. D. G. Roland Burlirsch and J. Schröder, editors, *Numerical Treatment of Differential Equations*, volume 631 of *Lecture Notes in Mathematics*, pages 133–146. Springer, New York, Berlin, Heidelberg, Tokyo, 1976. Proceedings, Oberwolfach, 1976.

[15] S. H. Paskov and J. Traub. Faster Valuation of Financial Derivatives. *Journal of portfolio management*, pages 113–120, 1995.

[16] I. M. Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Comput. Math. Math. Phys.*, 7:86–112, 1967.

[17] G. Stengle. Numerical methods for systems with measureable coefficients. *Appl. Math. Lett.*, 3:25–29, 1990.

[18] G. Stengle. Error analysis of a randomized numerical method. *Numer. Math.*, 70:119–128, 1995.

Reinhold Kainhofer
Robert F. Tichy
Department of Mathematics
Graz University of Technology
Steyrergasse 30
8010 Graz
Austria