Annales Univ. Sci. Budapest., Sect. Comp. 58 (2025) 125–140

MULTISTEP ALGORITHM TO SOLVE GENERAL RICCATI DIFFERENTIAL EQUATIONS BY WALSH POLYNOMIALS APPROACH

György Gát (Debrecen, Hungary)
János Márki-Zay and Rodolfo Toledo (Budapest, Hungary)

Communicated by László Szili
(Received 29 April 2025; accepted 5 July 2025)

Abstract. In this paper, we aim to introduce a procedure for the numerical solution of initial value problems associated with general Riccati-type differential equations. The core idea of the method is based on the approach developed by György Gát and Rodolfo Toledo, which has been successfully applied to the solution of linear differential equations. Our goal is to provide an iterative method that, starting from the initial point, constructs a Walsh polynomial that approximates the theoretical solution of the problem.

1. Introduction

The Walsh functions are widely used in the digital world. The main reason for this is that they form an orthonormal system that can take only two values. Recognizing this fact, in the 1970s several researchers began to study the applications of Walsh functions intensively, e.g., in telecommunications and signal processing (see e.g., [10, 12]). Corrington [5] was the first to develop a method for solving initial value problems for higher-order linear differential equations using Walsh functions. Corrington's method was quite complicated, left many details unclear, and required a significant amount of computation.

Key words and phrases: Fourier analysis, Walsh–Paley system, numerical solutions of differential equations, Riccati differential equations.
2010 Mathematics Subject Classification: 42C10, 65L05.
The first author was supported by the University of Debrecen Program for Scientific Publication grant no. PTP/0484/2024
https://doi.org/10.71352/ac.58.050725
First published online: 9 July 2025

In 1975, C. F. Chen and C. H. Hsiao further developed Corrington's method by providing a solution based on Walsh functions for first-order linear systems of differential equations. Their method was much simpler, easier to implement, and worked with matrices that can be generated quickly. Chen and Hsiao's method differed significantly from Corrington's method. Instead of successive approximations, the solution of an initial value problem was approximated by Walsh polynomials whose coefficients can be obtained from a system of linear equations, providing the exact solution (and not its derivative!). However, the method was developed exclusively for equations with constant coefficients. In that year, Chen and Hsiao published several articles in which they demonstrated the applicability of their method to solving various practical problems (see [1, 2, 3, 4]). The method is also adaptable for solving first-order linear partial differential equations using two-dimensional Walsh–Fourier series (see [14]).

The basic idea of Chen and Hsiao's method is to discretize the integral equation equivalent to the problem by replacing every function involved with the 2^n-th partial sum of its Walsh–Fourier series, including the integral function itself. The exact solution is approximated by a Walsh polynomial. In this way, a system of linear equations is obtained, whose unknowns are the coefficients of the Walsh polynomial. This Walsh polynomial will be an approximate solution to the exact solution of the problem. However, Chen and Hsiao did not address the analysis of the numerical solution. They did not investigate whether the proposed numerical solution exists or whether it properly converges to the exact solution of the problem. On the other hand, in order to achieve the desired accuracy, it is necessary to solve a very large system of linear equations consisting of $m2^n$ equations, where m is the number of equations in the problem, and 2^n is the number of Walsh functions used in the numerical solution. Moreover, after solving the system of equations, the Walsh polynomial must be constructed from the obtained coefficients, which is also a time- and computation-intensive task.

This was the reason why György Gát and Rodolfo Toledo began to work on the mathematical analysis of the method (see [7, 8]). They first dealt with the simplest case, namely the initial value problem

$$y' + ay = q(x), \qquad y(0) = \eta,$$

where $a, \eta \in \mathbb{R}$ are constants and $q\colon [0,1) \to \mathbb{R}$ is a continuous and integrable function. They proved that a numerical solution does not exist only if $2^{n+1} = -a$ holds, which can occur at most for a single value of n. Along with providing an error estimate, they were able to prove that the numerical solutions converge uniformly to the exact solution of the problem as $n \to \infty$. Furthermore, they developed a fast iterative procedure that directly produces the values of the Walsh polynomial without the need to solve systems of equations. After that, they proceeded to study initial value problems with non-constant coefficients and succeeded in generalizing the results obtained for the constant coefficient case. They proved that numerical solutions can be obtained except for, at most, finitely many values of n. Similarly, they succeeded in proving uniform convergence and in constructing an iterative procedure. These results were published in [9].

It is important to note that they aimed to address the problem under the most general conditions possible. This is reflected in the fact that the functions involved are not necessarily continuous on the closed interval $[0,1]$. The condition was that the functions be continuous on the interval $[0,1)$ and Lebesgue integrable. Success could be achieved under such conditions because the method works with integral functions, which are already continuous on the closed interval $[0,1]$. Furthermore, instead of approximating the derivative of the exact solution by step functions, as is done in other similar methods (see, e.g., [11]), they provided an approximation directly for the exact solution itself.

However, the solvability of the integral equation does not require the continuity of the functions appearing in the equation. It is known that if I is an interval and f is a continuous vector-valued function, then the initial value problem

$$y' = f(x,y), \qquad y(\xi) = \eta \qquad (x \in I)$$

is equivalent to the integral equation

$$y(x) = \eta + \int\limits_{\xi}^{x} f(x, y(x))\,dx \qquad (x \in I).$$

If f is not continuous in its first variable (x), but is continuous in its second variable (y), then the integral equation may have a non-differentiable solution, which of course cannot be a solution of the initial value problem. This is exactly what happens in the case of systems of linear differential equations when the functions involved are integrable on every closed subinterval of I since, in that case, the integral equation has a unique continuous solution on the entire interval I (see [6]). Recognizing this, Rodolfo Toledo [15] generalized the method to systems of first-order linear differential equations as well.

The successes achieved in the case of linear differential equations motivate the development of a numerical method based on Walsh functions for other types of differential equations. A natural starting point is to consider initial value problems related to Riccati-type differential equations. This is also the aim of the present paper.

2. Preliminaries

2.1. Notation and objectives

Consider the following initial value problem for a Riccati differential equation:

$$\begin{aligned} y' + p_1(x)y + p_2(x)y^2 &= q(x), \\ y(0) &= \eta, \end{aligned} \tag{2.1}$$

where $p_1, p_2, q\colon [0,1) \to \mathbb{R}$ are continuous functions, and $\eta \in \mathbb{R}$. We further assume that the functions p_1, p_2, and q are Lebesgue integrable, that is,

$$\int_0^1 |p_1(x)|\,dx < \infty, \quad \int_0^1 |p_2(x)|\,dx < \infty, \quad \text{and} \quad \int_0^1 |q(x)|\,dx < \infty.$$

According to the Picard–Lindelöf theorem, there exists a number $0 < \beta \le 1$ such that (2.1) admits a unique solution y on the interval $[0,\beta)$. However, in general, this solution cannot be extended beyond β.

Our objective is to develop a numerical procedure to approximate the solution using Walsh polynomials of the form

$$\overline{y}_n(x) = \sum_{k=0}^{2^n-1} c_k w_k(x), \tag{2.2}$$

where ω_i denotes the ith Walsh function ordered in the Paley sense. A similar approach was proposed by Gát and Toledo in [9] for first-order linear differential equations.

To this end, we consider the equivalent integral equation

$$y(x) = \eta + \int_0^x \big(q(t) - p_1(t)y(t) - p_2(t)y^2(t)\big)\,dt.$$

We discretize this equation by replacing each function with the 2^n-th partial sum of its Walsh–Fourier series, including the integral term. That is, we seek Walsh polynomials $\overline{y}_n$ satisfying the relation

$$\overline{y}_n(x) = \eta + S_{2^n} \int_0^x \big(S_{2^n} q(t) - S_{2^n} p_1(t)\,\overline{y}_n(t) - S_{2^n} p_2(t)\,\overline{y}_n^2(t)\big)\,dt \tag{2.3}$$

for all $0 \le x < 1$, where $S_{2^n} f$ denotes the 2^n-th partial sum of the Walsh–Fourier series of the integrable function f.

It is important to clarify that the expression

$$S_{2^n}\left(\int_0^{\cdot} f(t)\,dt\right)(x)$$

refers to the 2^n-th partial sum of the Walsh–Fourier series of the integral function

$$\int_0^x f(t)\,dt \qquad (x \in [0,1).$$

2.2. The 2^n-th partial sums of the Walsh–Fourier series

We begin by introducing the concept of the Walsh–Paley functions. Every integer $n \in \mathbb{N}$ admits a unique dyadic expansion:

$$n = \sum_{k=0}^{\infty} n_k 2^k,$$

where each $n_k \in \{0,1\}$. This sequence $(n_0, n_1, \dots)$ is called the dyadic expansion of n. Similarly, any real number $x \in [0,1)$ has a dyadic expansion of the form

$$x = \sum_{k=0}^{\infty} \frac{x_k}{2^{k+1}},$$

where again $x_k \in \{0,1\}$ for all $k \in \mathbb{N}$. This expansion is not unique when x is a dyadic rational, i.e., when $x = \frac{i}{2^k}$ for some $i, k \in \mathbb{N}$ with $0 \le i < 2^k$. In such cases, we choose the representation that terminates with zeros. For two numbers $x, y \in [0,1)$ with dyadic expansions $(x_0, x_1, \dots)$ and $(y_0, y_1, \dots)$, respectively, we define their dyadic sum by

$$x \dot{+} y := \sum_{k=0}^{\infty} |x_k - y_k| 2^{-(k+1)}.$$

The Walsh functions are finite products of the so-called Rademacher functions:

$$r_k(x) := (-1)^{x_k}, \qquad (x \in [0,1),\ k \in \mathbb{N}).$$

The Walsh–Paley system is defined by ordering the Walsh functions as follows:

$$w_n(x) := \prod_{k=0}^{\infty} r_k^{n_k}(x), \qquad (x \in [0,1),\ n \in \mathbb{N}),$$

where $n = \sum n_k 2^k$ is the dyadic expansion of n. The function w_n is referred to as the nth Walsh–Paley function or simply the nth Walsh function in Paley ordering. A Walsh polynomial is any finite linear combination of these functions.

The Walsh–Paley system forms an orthonormal system:

$$\int_0^1 w_n(x)w_m(x)\,dx = \begin{cases} 1, & \text{if } n = m, \\ 0, & \text{if } n \neq m. \end{cases}$$

Among other consequences, this orthonormality implies that two Walsh polynomials are identical (almost everywhere) if and only if their Walsh coefficients coincide.

Throughout this paper, we work with real-valued functions defined on $[0, 1)$. For any integrable function f satisfying

$$\int_0^1 |f(x)|\,dx < \infty,$$

the Walsh–Fourier coefficients and the corresponding partial sums are defined by

$$\widehat{f}_k := \int_0^1 f(x)w_k(x)\,dx, \qquad (k \in \mathbb{N}),$$

$$S_n f(x) := \sum_{k=0}^{n-1} \widehat{f}_k w_k(x), \qquad (n \in \mathbb{N},\ x \in [0, 1)).$$

It is particularly important that the 2^n-th partial sum admits the representation

$$S_{2^n} f(x) = 2^n \int_{I_n(x)} f(y)\,dy,$$

where

$$I_k(i) := \left[\frac{i-1}{2^k}, \frac{i}{2^k}\right), \qquad (i = 1, \ldots, 2^k),$$

are the dyadic intervals of length 2^{-k}, which contain the point x (see [13]).

For every integrable function f, the sequence $S_{2^n} f$ converges to f in the L^1 norm (see [13], p. 142), i.e.,

$$\lim_{n\to\infty} \int_0^1 |S_{2^n} f(x) - f(x)|\,dx = 0.$$

Continuity on $[0, 1)$ does not imply integrability. However, if a continuous function has a finite limit as $x \to 1^-$, it can be extended continuously to $[0, 1]$ and thus becomes integrable. In this case, one can show (see [13]) that $S_{2^n} f$ converges uniformly to f on $[0, 1)$.

The following lemma is crucial for the analysis of the numerical scheme developed in this paper. This lemma was proved in [9].

Lemma 2.1. *Let $f\colon [0,1) \to \mathbb{R}$ be constant on the dyadic intervals of length 2^{-n}, and suppose $x \in I_n(i)$ for some $i = 1, \ldots, 2^n$. Then*

$$S_{2^n}\left(\int\limits_0^{\cdot} f(t)\,dt\right)(x) = \frac{1}{2^n}\sum_{k=1}^{i-1} f\left(\frac{k-1}{2^n}\right) + \frac{1}{2^{n+1}} f\left(\frac{i-1}{2^n}\right).$$

3. Construction of the numerical solution

The method developed by György Gát and Rodolfo Toledo has two types of approaches:

1) From the discretized integral equation, we obtain a system of 2^n linear equations, whose solutions provide the coefficients of the desired Walsh polynomial (2.2).
2) The direct application of Lemma 2.1 to the discretized integral equation leads to an iterative procedure, which, starting from the initial value, directly yields the values of the desired Walsh polynomial (2.2) on the dyadic intervals.

The first approach was initially proposed by Chen and Hsiao for systems of linear differential equations with constant coefficients. However, due to its high computational cost, it is not suitable for obtaining high-precision numerical solutions. Therefore, in this paper, we will present the second approach, that is, we will work with the iterative method.

Now we return to the analysis of the discretized integral equation (2.3), that is,

$$\overline{y}_n(x) = \eta + S_{2^n}\int\limits_0^x \left(S_{2^n}q(t) - S_{2^n}p_1(t)\,\overline{y}_n(t) - S_{2^n}p_2(t)\,\overline{y}_n^2(t)\right) dt$$

Note that the functions appearing under the integral are constant on the dyadic intervals of length 2^{-n}, which allows us to apply Lemma 2.1 to rewrite the integral. Let $i = 1, 2, \ldots, 2^n$, and assume $\frac{i-1}{2^n} \le x < \frac{i}{2^n}$. Then, we have

$$\begin{aligned}
\overline{y}_n\left(\frac{i-1}{2^n}\right) &= \overline{y}_n(x) = \\
&= \eta + S_{2^n}\left(\int\limits_0^{\cdot} S_{2^n}q(t) - S_{2^n}p_1(t)\overline{y}_n(t) - S_{2^n}p_2(t)\overline{y}_n^2(t)\,dt\right)(x) = \\
&= \eta + \frac{1}{2^n}\sum_{k=1}^{i-1}\left(S_{2^n}q\left(\tfrac{k-1}{2^n}\right) - S_{2^n}p_1\left(\tfrac{k-1}{2^n}\right)\overline{y}_n\left(\tfrac{k-1}{2^n}\right) - S_{2^n}p_2\left(\tfrac{k-1}{2^n}\right)\overline{y}_n^2\left(\tfrac{k-1}{2^n}\right)\right) + \\
&\quad + \frac{1}{2^{n+1}}\left(S_{2^n}q\left(\tfrac{i-1}{2^n}\right) - S_{2^n}p_1\left(\tfrac{i-1}{2^n}\right)\overline{y}_n\left(\tfrac{i-1}{2^n}\right) - S_{2^n}p_2\left(\tfrac{i-1}{2^n}\right)\overline{y}_n^2\left(\tfrac{i-1}{2^n}\right)\right).
\end{aligned}$$

This means that for each step, one must solve a quadratic equation of the form

$$\frac{1}{2^{n+1}} S_{2^n} p_2\left(\tfrac{i-1}{2^n}\right) \overline{y}_n^2\left(\tfrac{i-1}{2^n}\right) + \left(1 + \frac{1}{2^{n+1}} S_{2^n} p_1\left(\tfrac{i-1}{2^n}\right)\right) \overline{y}_n\left(\tfrac{i-1}{2^n}\right) =$$
$$= \eta + \frac{1}{2^n} \sum_{k=1}^{i-1} \left(S_{2^n} q\left(\tfrac{k-1}{2^n}\right) - S_{2^n} p_1\left(\tfrac{k-1}{2^n}\right) \overline{y}_n\left(\tfrac{k-1}{2^n}\right) - S_{2^n} p_2\left(\tfrac{k-1}{2^n}\right) \overline{y}_n^2\left(\tfrac{k-1}{2^n}\right)\right) +$$
$$+ \frac{1}{2^{n+1}} S_{2^n} q\left(\tfrac{i-1}{2^n}\right),$$

in order to compute $\overline{y}_n\left(\frac{i-1}{2^n}\right)$. This is feasible since the right-hand side only depends on the values of $\overline{y}_n$ at points strictly less than $\frac{i-1}{2^n}$.

The algorithm proceeds from $i = 1$ up to $i = 2^n$, or until the quadratic equation becomes unsolvable. At the initial step ($i = 1$), the equation simplifies to

$$\text{(3.1)} \qquad \frac{1}{2^{n+1}} S_{2^n} p_2(0)\, \overline{y}_n^2(0) + \left(1 + \frac{1}{2^{n+1}} S_{2^n} p_1(0)\right) \overline{y}_n(0) = \eta + \frac{1}{2^{n+1}} S_{2^n} q(0),$$

which must be solved for $\overline{y}_n(0)$.

At each step, we solve a quadratic equation using the standard quadratic formula. However, it is essential to determine which root should be selected in the algorithm. Observe that at the initial step ($k = 1$), the "smaller" root of (3.1) is given by

$$\overline{y}_n(0) = \frac{-1 - \frac{1}{2^{n+1}} S_{2^n} p_1(0) - \sqrt{D}}{\frac{1}{2^n} S_{2^n} p_2(0)},$$

where

$$D := \left(1 + \frac{1}{2^{n+1}} S_{2^n} p_1(0)\right)^2 + \frac{4}{2^{n+1}} S_{2^n} p_2(0) \left(\eta + \frac{1}{2^{n+1}} S_{2^n} q(0)\right),$$

but this means that $\overline{y}_n(0)$ does not converge to η as $n \to \infty$, since it is asymptotically of the form $\left(\frac{-2}{0}\right)$.

For this reason, the algorithm consistently selects the "larger root" of the quadratic equation at each step.

4. Implementation of the iterative method in Matlab

4.1. Code implementation

The implementation of the algorithm in Matlab is straightforward. The function `odeRiccatiWalsh0` takes as input the value of `n`, the parameters of the Riccati differential equation (provided as function handles), and `eta`, which specifies the initial condition $y(0) = \eta$. The function returns three outputs: the starting points of the dyadic intervals of length 2^{-n}, the corresponding approximate values of the solution on these intervals, and a status flag. This third output is zero if the solution was successfully computed on all intervals; otherwise, it gives the index of the first interval where the computation failed. In such a case, the returned solution contains the value `Inf` from that interval onward.

```
function [t,yNum,szing]=odeRiccatiWalsh0(n,p1,p2,q,eta,atol,rtol)
% It solves numerically the intial value problem
% y'+p1(x)y+p2(x)y^2=q(x), y(0)=eta
% using Walsh functions, where the real functions p1, p2 and q ...
    are defined on the interval [0,1[.

% Input variables
% n: it sets the dimension of the solution
% p1: coefficient function of the linear term (function handle)
% p2: coefficient function of the quadratic term (function handle)
% q: free term (function handle)
% atol: absolute tolerance used only for numerical integration
% rtol: relative tolerance used only for numerical integration

% Auxiliary variables
dim = 2^n;
FSp1 = zeros(1,dim);
FSp2 = zeros(1,dim);
FSq = zeros(1,dim);

% Output variables
szing = 0;
t = 0:1/dim:1-1/dim;
yNum = zeros(1,dim);

% Computation of the integral means
for i = 1:dim
    FSp1(i) = integral(p1NumF,(i-1)/dim,i/dim, ...
        'ArrayValued',true,'AbsTol',atol,'RelTol',rtol);
    FSp2(i) = integral(p2NumF,(i-1)/dim,i/dim, ...
        'ArrayValued',true, 'AbsTol',atol,'RelTol',rtol);
    FSq(i) = integral(qNumF,(i-1)/dim,i/dim, ...
        'ArrayValued',true,'AbsTol',atol,'RelTol',rtol);
end

% Iterative computation of the approximated solution
% Calculation stops when the quadratic equation has no real ...
    root (D < 0)
szum = eta;
for i = 1:dim
    a = FSp2(i)/2;
    b = 1+FSp1(i)/2;
    c = szum+FSq(i)/2;
    D = b^2+4*a*c;
    if a == 0
        yNum(i) = c/b;
    elseif D ≥ 0
```

```
           yNum(i) = (-b+sqrt(D))/(2*a);
        else
           for j = i:dim
               yNum(j) =  Inf;
           end
           szing = i;
           break
        end
        szum = szum+FSq(i)-FSp1(i)*yNum(i)-FSp2(i)*yNum(i)^2;
    end
    end
```

4.2. The modified solution

Most numerical methods for solving differential equations determine the value of the solution only at a few points and apply linear interpolation elsewhere. The result is a piecewise linear solution. However, Walsh polynomials are step functions, which makes our developed method difficult to compare with other well-known numerical methods. Recognizing this, Rodolfo Toledo (see [15]) proposed the following modification: We retain the values of the resulting Walsh polynomial only at the midpoints of the intervals; we also add the initial point (ξ, η) and apply linear interpolation through these points. At the right half of the last interval, the interpolation is extended accordingly.

In this way, we obtain a piecewise linear solution, which we refer to as the modified solution. As shown in [15], this modification significantly improves the accuracy of the approximation for linear differential equations. Empirical evidence suggests that in the original Walsh polynomial solution, the difference between the exact and numerical solutions is typically halved when the value of n is increased by one. In contrast, for the modified solution, this difference is reduced to nearly a quarter. As the next section will show, similar behavior can be observed in the case of Riccati differential equations as well.

4.3. Examples

In this section, we present some examples where classical numerical methods either do not work or are not sufficiently effective.

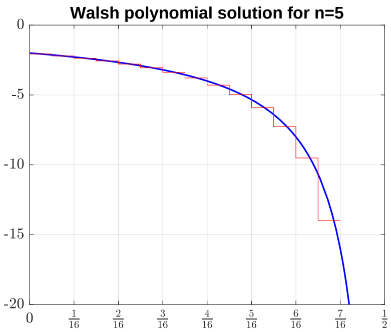
Consider the following problem:

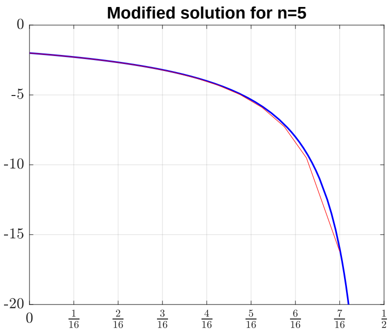
$$y' + y^2 = 0, \qquad y(0) = -2$$

on the interval $[0, 1)$. The exact solution of this problem is known:

$$y(x) = \frac{1}{x - \frac{1}{2}} \qquad (0 \leq x < \tfrac{1}{2}).$$

The solution has a singularity at $x = 1/2$. The next figures below illustrate the solutions obtained using the original and the modified methods for $n = 5$.





We observe that no value was obtained for the last interval $[7/16, 1/2)$. This is not an isolated case. We ran the algorithm up to $n = 10$ and experienced the same phenomenon, namely that no value was obtained on the last interval of length 2^{1-n} because in that interval, we must solve a quadratic equation whose discriminant is negative.

To examine the accuracy, we computed the maximum absolute difference between the exact and the numerical solutions over certain intervals. This is shown in the table below for Walsh polynomial solutions. It can be observed that the difference roughly halves when n is incremented by one. Near $1/2$, the absolute difference increases since the exact solution is unbounded in that region.

n	$\left[0, \frac{1}{8}\right)$	$\left[\frac{1}{8}, \frac{2}{8}\right)$	$\left[\frac{2}{8}, \frac{3}{8}\right)$	$\left[\frac{3}{8}, \frac{7}{16}\right)$	$\left[\frac{7}{16}, \frac{15}{32}\right)$	$\left[\frac{15}{32}, \frac{31}{64}\right)$	$\left[\frac{31}{64}, \frac{63}{128}\right)$	$\left[\frac{63}{128}, \frac{127}{256}\right)$
3	3.43e-01	7.48e-01	Inf	Inf	Inf	Inf	Inf	Inf
4	1.93e-01	4.22e-01	1.60e+00	Inf	Inf	Inf	Inf	Inf
5	1.03e-01	2.28e-01	8.71e-01	3.30e+00	Inf	Inf	Inf	Inf
6	5.35e-02	1.19e-01	4.64e-01	1.76e+00	6.72e+00	Inf	Inf	Inf
7	2.79e-02	6.10e-02	2.40e-01	9.35e-01	3.56e+00	1.35e+01	Inf	Inf
8	1.37e-02	3.08e-02	1.22e-01	4.83e-01	1.87e+00	7.15e+00	2.72e+01	Inf
9	6.91e-03	1.55e-02	6.18e-02	2.45e-01	9.68e-01	3.76e+00	1.43e+01	5.45e+01
10	3.46e-03	7.78e-03	3.10e-02	1.23e-01	4.91e-01	1.93e+00	7.53e+00	2.86e+01

Next, we show the same difference, but now with respect to the modified solution. In the table below, it can be seen that the difference now decreases roughly to a quarter when n is incremented by one.

n	$\left[0, \frac{1}{8}\right)$	$\left[\frac{1}{8}, \frac{2}{8}\right)$	$\left[\frac{2}{8}, \frac{3}{8}\right)$	$\left[\frac{3}{8}, \frac{7}{16}\right)$	$\left[\frac{7}{16}, \frac{15}{32}\right)$	$\left[\frac{15}{32}, \frac{31}{64}\right)$	$\left[\frac{31}{64}, \frac{63}{128}\right)$	$\left[\frac{63}{128}, \frac{127}{256}\right)$
3	2.12e-01	2.38e-01	Inf	Inf	Inf	Inf	Inf	Inf
4	4.38e-02	1.77e-01	5.68e-01	Inf	Inf	Inf	Inf	Inf
5	1.05e-02	4.02e-02	3.90e-01	1.23e+00	Inf	Inf	Inf	Inf
6	2.61e-03	9.83e-03	8.84e-02	8.17e-01	2.55e+00	Inf	Inf	Inf
7	6.51e-04	2.44e-03	2.16e-02	1.85e-01	1.67e+00	5.21e+00	Inf	Inf
8	1.62e-04	6.10e-04	5.38e-03	4.52e-02	3.78e-01	3.38e+00	1.05e+01	Inf
9	4.06e-05	1.52e-04	1.34e-03	1.12e-02	9.24e-02	7.64e-01	6.79e+00	2.11e+01
10	1.01e-05	3.81e-05	3.35e-04	2.80e-03	2.29e-02	1.86e-01	1.53e+00	1.36e+01

In summary, we have observed the interesting phenomenon that, in this problem, our procedure generates solutions whose domain expands and gradually gets ”closer” to the domain of the exact solution, while their values tend to approach the values of the exact solution.

Let us examine another stiff problem:

$$y' + 100y^2 = 100x^2 + 1, \qquad y(0) = 1$$

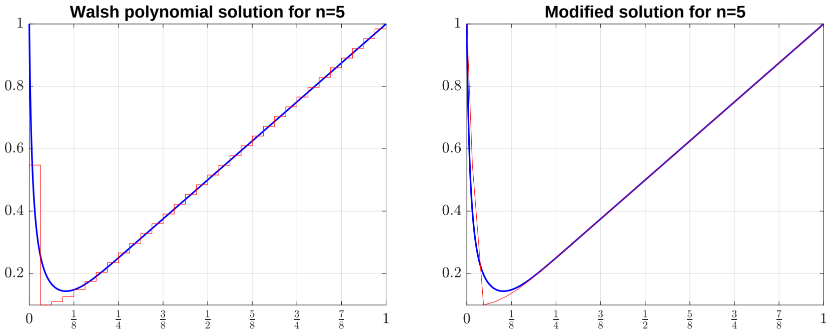
on the interval $[0, 1)$, whose exact solution is given by

$$y(x) = x + \frac{e^{-100x^2}}{5\sqrt{\pi}\,\mathrm{erf}(10x) + 1} \qquad (x \in [0, 1)),$$

where erf denotes the error function, defined by

$$\mathrm{erf}(x) := \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt.$$

The next figures illustrate the results obtained using the original iterative algorithm and its modified version for $n = 5$.


Walsh polynomial solution for n=5

Modified solution for n=5

This example is particularly interesting because the exact solution decreases abruptly in the right neighborhood of 0. As a result, our solution exhibits a notable deviation from the exact solution in this region, and the absolute error decreases slowly as n increases. However, despite the fact that the iteration starts from 0, it is capable of correcting itself and providing appropriate approximations over most of the interval. Most methods developed for non-stiff problems were not able to achieve this. The following table shows the maximum absolute difference over certain subintervals for the modified solution. We attempted to subdivide the ”problematic” region into smaller intervals. As we move away from this region, the deviation tends to decrease by roughly a factor of four with each increment in n.

n	$\left[0, \frac{1}{128}\right)$	$\left[\frac{1}{128}, \frac{1}{64}\right)$	$\left[\frac{1}{64}, \frac{1}{32}\right)$	$\left[\frac{1}{32}, \frac{1}{16}\right)$	$\left[\frac{1}{16}, \frac{1}{8}\right)$	$\left[\frac{1}{8}, \frac{1}{4}\right)$	$\left[\frac{1}{4}, \frac{1}{2}\right)$	$\left[\frac{1}{2}, 1\right)$
3	3.51e-01	4.38e-01	4.47e-01	4.16e-01	1.79e-01	2.10e-01	1.01e-01	1.96e-03
4	2.93e-01	3.24e-01	3.20e-01	1.80e-01	2.35e-01	1.71e-01	7.46e-03	3.40e-04
5	2.07e-01	2.08e-01	1.49e-01	9.74e-02	6.19e-02	1.09e-02	1.34e-04	8.31e-05
6	1.14e-01	1.03e-01	7.15e-02	1.91e-02	7.61e-03	1.31e-03	3.38e-05	2.07e-05
7	5.98e-02	5.09e-02	6.49e-03	4.33e-03	1.71e-03	3.21e-04	8.48e-06	5.19e-06
8	2.80e-02	8.84e-03	1.43e-03	1.07e-03	4.34e-04	8.20e-05	2.12e-06	1.29e-06
9	1.06e-02	2.07e-03	3.51e-04	2.73e-04	1.10e-04	2.09e-05	5.30e-07	3.24e-07
10	3.46e-03	5.12e-04	8.74e-05	6.89e-05	2.79e-05	5.28e-06	1.32e-07	8.11e-08

Finally, we consider a problem where the functions involved are unbounded on the interval $[0, 1)$, yet remain integrable. Let us examine the following problem:

$$y' - \frac{y^2}{\sqrt{1-x}} + \frac{y}{\sqrt{1-x}} = 2x + \frac{x^2-1}{\sqrt{1-x}} - \frac{(x^2-1)^2}{\sqrt{1-x}}, \qquad y(0) = -1$$

defined on the interval $[0, 1)$, whose exact solution is simple:

$$y(x) = x^2 - 1 \qquad (x \in [0, 1)).$$

In the case of linear differential equations, for problems of this type, the method based on Walsh functions worked well. The same holds true in the present case, as shown in the table below. It presents the maximum absolute difference on dyadic subintervals of length 1/8 with respect to the modified solution.

n	$\left[0, \frac{1}{8}\right)$	$\left[\frac{1}{8}, \frac{2}{8}\right)$	$\left[\frac{2}{8}, \frac{3}{8}\right)$	$\left[\frac{3}{8}, \frac{4}{8}\right)$	$\left[\frac{4}{8}, \frac{5}{8}\right)$	$\left[\frac{5}{8}, \frac{6}{8}\right)$	$\left[\frac{6}{8}, \frac{7}{8}\right)$	$\left[\frac{7}{8}, 1\right)$
3	7.07e-03	7.06e-03	6.52e-03	6.22e-03	6.16e-03	6.58e-03	1.20e-02	1.37e-02
4	1.84e-03	1.75e-03	1.62e-03	1.54e-03	1.53e-03	1.61e-03	1.94e-03	5.25e-03
5	4.74e-04	4.38e-04	4.05e-04	3.87e-04	3.82e-04	4.02e-04	4.73e-04	2.11e-03
6	1.20e-04	1.09e-04	1.01e-04	9.67e-05	9.57e-05	1.00e-04	1.17e-04	8.79e-04
7	3.02e-05	2.73e-05	2.53e-05	2.41e-05	2.39e-05	2.51e-05	2.93e-05	3.49e-04
8	7.59e-06	6.84e-06	6.33e-06	6.04e-06	5.98e-06	6.28e-06	7.34e-06	1.34e-04
9	1.90e-06	1.71e-06	1.58e-06	1.51e-06	1.49e-06	1.57e-06	1.83e-06	5.01e-05
10	4.76e-07	4.27e-07	3.95e-07	3.77e-07	3.73e-07	3.92e-07	4.58e-07	1.84e-05

5. Conclusion

The iterative procedure based on Walsh functions, developed by György Gát and Rodolfo Toledo for solving initial value problems associated with linear differential equations with non-constant coefficients, can be generalized to handle general Riccati-type differential equations as well. The method works under similar general conditions, even when the coefficient functions are unbounded.

The procedure is capable of dealing with singularities in the exact solutions by progressively approaching the domain of definition of the exact solution as the value of n increases. Regarding the accuracy of the numerical solutions, the behavior is similar to that observed in the case of linear differential equations: if the exact solution is not too steep, then the error of the Walsh polynomial solution is roughly halved, and that of the modified solution is reduced to approximately a quarter when n is increased by one.

Based on these results, we recommend a complete mathematical analysis of the method.

References

[1] **Chen, C.F. and C.H. Hsiao,** A Walsh series direct method for solving variational problems, *Journal of the Franklin Institute*, **300(4)** (1975), 265–280.
https://doi.org/10.1016/0016-0032(75)90199-4

[2] **Chen, C.F. and C.H. Hsiao,** Design of piecewise constant gains for optimal control via Walsh functions, *IEEE Transactions on Automatic Control*, **20(5)** (1975), 596–603.
https://doi.org/10.1109/TAC.1975.1101057

[3] **Chen, C.F. and C.H. Hsiao,** Time-domain synthesis via Walsh functions. *Proceedings of the Institution of Electrical Engineers*, **122(5)** (1975), 565–570.
https://doi.org/10.1049/piee.1975.0155

[4] **Chen, C.F. and C.H. Hsiao,** Walsh series analysis in optimal control, *International Journal of Control*, **21(6)** (1975), 881–897.
https://doi.org/10.1080/00207177508922043

[5] **Corrington, M.,** Solution of differential and integral equations with Walsh functions, *IEEE Transactions on Circuit Theory*, **20(5)** (1973), 470–476.
https://doi.org/10.1109/TCT.1973.1083748

[6] **Filippov, A.F.,** *Differential equations with discontinuous righthand sides*, Kluwer Academic Publishers Group, 1988.
https://doi.org/10.1007/978-94-015-7793-9

[7] **Gát, G. and R. Toledo,** A numerical method for solving linear differential equations via Walsh functions, In *Advances in Information Science and Applications*, **2** (2014), 334–339. (Proceedings of the 18th International Conference on Computers (part of CSCC 2014), Santorini Island, Greece, July 17–21, 2014)

[8] **Gát, G. and R. Toledo,** Estimating the error of the numerical solution of linear differential equations with constant coefficients via Walsh polynomials. *Acta Math. Acad. Paedagog. Nyházi. (N.S.)*, **31(2)** (2015), 309–330.

[9] **Gát, G. and R. Toledo,** Numerical solution of linear differential equations by Walsh polynomials approach, *Studia Scientiarum Mathematicarum Hungarica*, **57(2)** (2020), 217–254.
https://doi.org/10.1556/012.2020.57.2.1459

[10] **Gibbs, J.E. and H.A. Gebbie,** Application of Walsh Functions to Transform Spectroscopy, *Nature*, **224** (1969), 1012–1013.
https://doi.org/10.1038/2241012a0

[11] **Lukomskii, D.S., S.F. Lukomskii and P.A. Terekhin,** Solution of Cauchy problem for equation first order via Haar functions, *Izv. Saratov Univ. (N.S.), Ser. Math. Mech. Inform.*, **16** (2016), 151–159.
https://doi.org/10.18500/1816-9791-2016-16-2-151-159

[12] **Rao, G.P.,** *Piecewise constant orthogonal functions and their application to systems and control.*, volume 55. Springer, Cham, 1983.
https://doi.org/10.1007/BFb0041228

[13] **Schipp, F., W.R. Wade and P. Simon,** *Walsh series. An introduction to dyadic harmonic analysis*, Adam Hilger, Bristol and New York, 1990.

[14] **Shih, Y.P. and J.Y. Han,** Double Walsh series solution of first-order partial differential equations, *International Journal of Systems Science*, **9(5)** (1978), 569–578.
https://doi.org/10.1080/00207727808941720

[15] **Toledo, R.,** Solving systems of linear differential equations by Walsh polynomials approach, *Annales Univ. Sci. Budapest., Sect. Comp.*, **52** (2021), 313–348.
https://doi.org/10.71352/ac.52.313

György Gát
https://orcid.org/0000-0003-0907-6821
University of Debrecen
Institute of Mathematics
H-4032 Debrecen, Egyetem tér 1
Hungary
gat.gyorgy@science.unideb.hu

János Márki-Zay
ELTE Eötvös Loránd University
Faculty of Informatics
Department of Numerical Analysis
H-1117 Budapest, Pázmány Péter sétány 1/C
Hungary
mzjanos2002@gmail.com

Rodolfo Toledo
https://orcid.org/0000-0002-9590-2538
ELTE Eötvös Loránd University
Faculty of Informatics
Department of Numerical Analysis
H-1117 Budapest, Pázmány Péter sétány 1/C
Hungary
toledo@inf.elte.hu