A hybrid classical-quantum algorithm for solution of nonlinear ordinary differential equations

Alok Shukla [*1] and Prakash Vedula [†2]

[1]Ahmedabad University, India
[2]University of Oklahoma, USA

December 2, 2021

Abstract

A hybrid classical-quantum approach for the solution of nonlinear ordinary differential equations using Walsh-Hadamard basis functions is proposed. Central to this hybrid approach is the computation of the Walsh-Hadamard transform of arbitrary vectors, which is enabled in our framework using quantum Hadamard gates along with state preparation, shifting, scaling, and measurement operations. It is estimated that the proposed hybrid classical-quantum approach for the Walsh-Hadamard transform of an input vector of size N results in a considerably lower computational complexity ($\mathcal{O}(N)$ operations) compared to the Fast Walsh-Hadamard transform ($\mathcal{O}(N \log_2(N))$ operations). This benefit will also be relevant in the context of the proposed hybrid classical-quantum approach for the solution of nonlinear differential equations. Comparisons of results corresponding to the proposed hybrid classical-quantum approach and a purely classical approach for the solution of nonlinear differential equations (for cases involving one and two dependent variables) were found to be satisfactory. Some new perspectives relevant to the natural ordering of Walsh functions (in the context of both classical and hybrid approaches for the solution of nonlinear differential equations) and representation theory of finite groups are also presented here.

1 Introduction

Ordinary differential equations (ODEs) are often used for mathematical descriptions of models in many engineering and science (including those relevant to physical, biological, and social sciences) disciplines. As exact solutions to nonlinear ODEs are generally not available, various numerical methods (e.g. single-step, multi-step, finite difference, finite element, and spectral methods) are often used to obtain approximate solutions [7, 25]. The selection of appropriate numerical methods for the solution of nonlinear differential equations is guided by considerations of accuracy, numerical stability, stiffness, memory constraints and execution time.

Spectral or pseudospectral methods are often preferred for high-order accuracy [6]. In these methods, the solution is expressed as a linear combination of certain (global) basis functions and the coefficients of

---

[*]alok.shukla@ahduni.edu.in
[†]pvedula@ou.edu

these basis functions are optimally selected based on the underlying differential equation. These methods can be shown to have exponential convergence. They are often more accurate and consume less memory compared to finite difference or finite element methods.

While many different choices for orthogonal basis functions can be considered, we consider Walsh functions [26, 2] as basis functions for the solution of ordinary differential equations. Walsh functions and associated transforms have been found to be useful in many contexts relevant to signal processing [27], image compression [16], cryptography [21], solution of non-linear differential equations [3, 1], solution of variational problems [8] and solution of partial differential equations relevant to fluid dynamics (with discontinuities or shocks) [12, 13, 14]. Our choice of basis functions is motivated not only due to important properties of Walsh functions but also due to close the natural connection between the Walsh-Hadamard transform [2] and the Hadamard gate [22] that is widely used in many quantum circuits and algorithms.

Walsh functions, which are square-wave like functions and have values of ±1 at any point in the domain of interest, form a complete set of orthogonal basis functions. Unlike many other orthogonal basis functions, Walsh functions are closed under multiplication (e.g. product of two Walsh functions results in a function that is contained in the basis set). As Walsh basis functions are composed of square-wave like functions, they are well suited to represent discontinuous functions and/or rapidly changing functions. Coefficients of the Walsh basis functions can be obtained from a data vector in the physical domain using a discrete Walsh-Hadamard transform (matrix). It can be shown [17] that the discrete Walsh-Hadamard transform when applied to a 2^n one-dimensional data is equivalent to the discrete $n-$dimensional Fourier transform applied the same data arranged on a binary $n-$cube. In contrast to many other transforms, the discrete Walsh-Hadamard transform can be computed by addition and subtraction operations alone and without the need for multiplication. This feature enables fast and accurate computation of the discrete Walsh-Hadamard transform. Note that the computational complexities associated with basic implementations of addition and multiplication operations of two $p-$digit numbers can be estimated to be $\mathcal{O}(p)$ and $\mathcal{O}(p^2)$ respectively. While a basic implementation of the discrete Walsh-Hadamard transform has a computational complexity of $\mathcal{O}(N^2)$ (where N denotes the elements in the data vector), efficient algorithms similar to those used in the Fast Fourier Transform (FFT) exist for computation of Fast Walsh-Hadamard transform that has a complexity of $\mathcal{O}(N \log N)$.

Walsh functions can be ordered in many ways and commonly used orderings include the sequency ordering (according to the number of zero-crossings), the natural ordering (based on Hadamard matrices generated by Kronecker products) and the dyadic ordering [11]. While the matrix rows for the transform matrices corresponding to these orderings can be interchanged via the use of permutation matrices, more efficient approaches exist for such purposes. For instance, conversion from the natural to the sequency ordering can be accomplished via a combination of bit reversal and Gray code to binary conversion.

In the literature, many treatments of Walsh basis functions applied to the solution of differential equations have focused on the sequency ordering. In contrast, in this work, we consider the natural ordering of Walsh functions and present results (including integration matrices and associated inverses) based on this ordering. For convenience, we will refer to Walsh basis functions considered in the natural ordering (based on Hadamard matrices) as Walsh-Hadamard (WH) basis functions.

Our preference for this ordering is based on the insight that the transform matrix for this ordering corresponds to the Hadamard matrix describing quantum Hadamard gate commonly used in many quantum circuits and algorithms. This insight provides further motivation to explore the possibility of making use of quantum Hadamard gates for fast computation of the Walsh-Hadamard transform which is a key step in

the solution of differential equations based on Walsh functions.

Quantum algorithms were previously proposed for the solution of linear ordinary differential equations [9, 5, 4] and nonlinear ordinary differential equations [18, 20, 19]. It is worth noting that development of quantum algorithms in the latter category is particularly challenging owing to the linearity of quantum mechanics. Limitations of previous quantum algorithms for the solution of nonlinear differential equations can be associated with poor scaling (as resources needed increase exponentially [18] or quadratically [20] with integration time), restricted sparsity structure [20], dissipation, restricted nonlinearities or approximation errors associated with Carleman linearization [19], to name a few. In this paper, we propose a hybrid classical-quantum approach for the solution of nonlinear ordinary differential equations (ODEs) using Walsh-Hadamard basis functions. We present new perspectives for the solution of nonlinear ODEs for both classical and hybrid approaches based on the natural ordering of Walsh functions. In this context, we also explore some theoretical insights on Walsh-Hadamard transform arising from the character theory of finite groups. While Hadamard gates are commonly used in many quantum circuits and algorithms and can be used to naturally obtain the Hadamard transform (under certain conditions), there are challenges involved in extracting useful classical information particularly due to the ambiguity associated with the global phase while carrying out measurements. We address these challenges via a novel hybrid classical quantum approach for computation of Walsh-Hadamard transform for arbitrary input vectors. This approach makes use of the special structure of the Walsh-Hadamard transform and it involves a combination of shifting, scaling and measurement operations, along with state preparation and use of quantum Hadamard gates. The advantage of our hybrid approach is a considerably lower computational complexity ($\mathcal{O}(N)$) in comparison to classical Fast Walsh-Hadamard transform ($\mathcal{O}(N \log_2 N)$). This speedup can also be harnessed for the solution of nonlinear ODEs. The proposed hybrid classical quantum approaches for computation of Walsh-Hadamard transform (using quantum Hadamard gates) and solutions of nonlinear ordinary differential equations were successfully implemented and tested using the simulated environment of Qiskit (IBM's open source quantum computing platform).

The rest of this paper is organized as follows. In section 2, we present the definition, construction and properties of Walsh-Hadamard basis functions. The relation between Walsh functions and the character theory of finite groups is presented in section 3. Classical and quantum implementation of the Walsh-Hadamard transform are discussed in sections 3.3 and 3.4. Our proposal for a hybrid classical-quantum approach for Walsh-Hadamard transform is presented in section 4 and the associated computational complexity is discussed in section 4.1. This approach is central to the solution of nonlinear ordinary differential equations using Walsh-Hadamard basis functions. Some basic concepts relevant to function representation, integration and differentiation using Walsh-Hadamard basis functions are presented in sections 5, 5.1 and 5.2. The proposed hybrid quantum-classical approach for the solution of nonlinear ordinary differential equations is presented in section 6. A couple of computational examples based on this approach are presented in section 7 and conclusions are summarized in section 8.

2   Walsh Functions

Let $N = 2^n$ be a positive integer. For $j = 0, 1, 2, \cdots N - 1$, the Walsh functions are defined as follows

$$W_0(x) = 1 \quad \text{for } 0 \leq x \leq 1, \tag{2.1}$$

3

$$W_{2j}(x) = W_j(2x) + (-1)^j W_j(2x-1), \qquad (2.2)$$

$$W_{2j+1}(x) = W_j(2x) - (-1)^j W_j(2x-1), \qquad (2.3)$$

$$W_j(x) = 0 \quad \text{for } x < 0 \text{ and } x > 1. \qquad (2.4)$$

For $N = 8$ the Walsh functions are shown in Figure 1. It is clear that the Walsh functions take the values 1 or -1 on the interval $[0,1]$ and they are 0 everywhere else. The Walsh functions shown in Figure 1 are in the so-called *sequency order*. In sequency ordering the number of sign changes (or zero crossings) for the Walsh functions increase as the orders of the functions increase. The Walsh functions with an even number of sign changes are symmetric about the line $x = 1/2$ and the Walsh functions with odd numbers of sign changes are anti-symmetric about the line $x = 1/2$. In the literature, analogous to the trigonometric sine and cosine functions the $sal_j(x)$ and $cal_j(x)$ functions are defined as given below.

$$sal_j(x) = W_{2j-1}(x), \qquad (2.5)$$

$$cal_j(x) = W_{2j}(x). \qquad (2.6)$$

It can be shown that Walsh functions are orthonormal.

$$\int_{-\infty}^{\infty} W_m(x) W_n(x) \, dx = \begin{cases} N, & \text{if } m = n, \\ 0, & \text{if } m \neq n. \end{cases} \qquad (2.7)$$

(a) $W_0(x)$ (b) $W_1(x)$

(c) $W_2(x)$ (d) $W_3(x)$

(e) $W_4(x)$ (f) $W_5(x)$
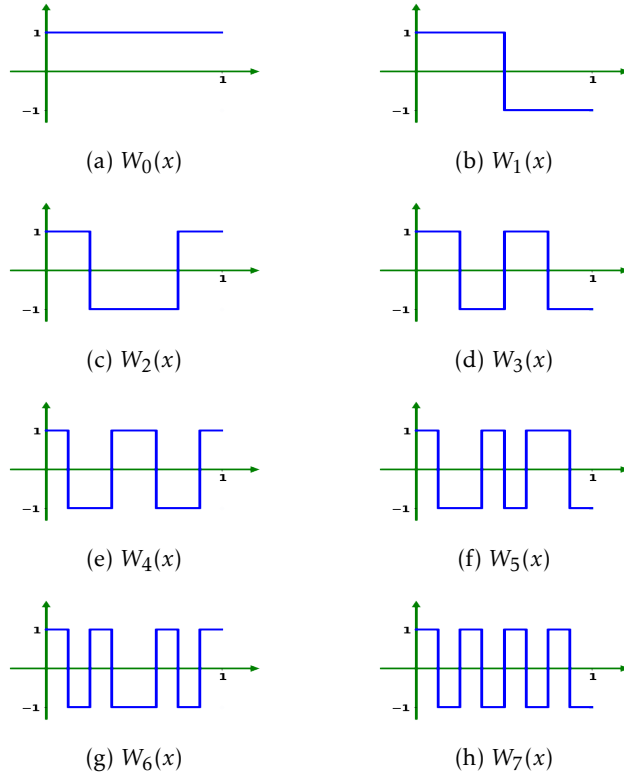
(g) $W_6(x)$ (h) $W_7(x)$

Figure 1: Walsh functions in the sequency ordering for $N = 8$.

We refer the readers to [2] for alternate definitions and properties of Walsh functions. Walsh functions can also be interpreted using the character theory of finite groups. Perhaps, this provides a conceptually cleaner approach to define equivalent of Walsh functions up to ordering. We will define Walsh functions using this approach in Section 3. In the rest of the paper, we will not use the sequency ordering of Walsh functions, instead, we will use the natural Hadamard ordering as used in the quantum computing community. Of course, it is easy to go from one ordering to another in applications.

3 Walsh Functions and the character theory of finite groups

We will first discuss some well-known results related to the Fourier analysis on finite abelian groups. We will work with a general finite abelian group in the beginning, and later specialize to $(\mathbb{Z}/2\mathbb{Z})^n$, leading to the theory of Walsh-Hadamard Transform. All the results in this section can be found in any standard textbook covering the representation theory of finite groups (e.g., [23], [10], [24]). While the character theory of finite groups is well-known its connections with Walsh-Hadamard transform is not adequately explored in the literature (to the best of our knowledge). Here we will explore these connections further and we will show that Walsh functions can be interpreted in terms of characters of certain abelian groups.

3.1 Fourier Analysis on Finite Groups

Let G be a finite group. We denote the vector space of all complex valued functions on G by $L(G)$, i.e.,

$$L(G) = \{ f \mid f : G \longrightarrow \mathbb{C} \}. \tag{3.1}$$

The addition and scalar multiplication in $L(G)$ are given by

$$(f_1 + f_2)(g) = f_1(g) + f_2(g) \quad \text{and} \quad (cf)(g) = cf(g), \quad \text{for } f_1, f_2 \in L(G), \text{ and } c \in \mathbb{C}. \tag{3.2}$$

The space $L(G)$ is an inner product space with the inner product

$$\langle f_1, f_2 \rangle = \frac{1}{|G|} \sum_{g \in G} f_1(g) \overline{f_2(g)}. \tag{3.3}$$

Next we recall some basic facts related to the representation theory of finite groups. Let $\phi : G \longrightarrow \mathrm{GL}(V)$ be a representation of G, where V is a finite dimensional vector space. It means each element $g \in G$ gives rise to a linear transformation $\phi(g)$ of the vector space V. The representation ϕ of the group G can be thought of the action of G on the vector space V, such that $\phi(gh) \cdot v = \phi(g) \cdot (\phi(h) \cdot v)$ for all $g, h \in G$ and $v \in V$, and $e \cdot v = v$ for the identity e of G. The dimension of V is called the dimension of ϕ. A subspace U of V is called G-invariant if $\phi(g) \cdot u \in U$ for all $u \in U$ and $g \in G$. A non-zero representation ϕ is said to be *irreducible* if $\{0\}$ and V are the only G-invariant subspaces of V. Two n-dimensional representations ϕ and ψ are called *equivalent representations* if there exists an invertible matrix $T \in \mathrm{GL}_n(\mathbb{C})$ such that $\phi(g) = T \psi(g) T^{-1}$ for every $g \in G$. If ϕ and ψ are equivalent representations then we write $\phi \sim \psi$. A finite dimensional representation ϕ is called unitary if $\langle \phi(g) \cdot u, \phi(g) \cdot v \rangle = \langle u, v \rangle$ for all $u, v \in V$ and $g \in G$. It is a fact that every representation of a finite group G is equivalent to a unitary representation. It is easy to see that a one dimensional unitary representation is a homomorphism $\phi : G \longrightarrow S^1$, where $S^1 = \{ z \in \mathbb{C} \mid |z| = 1 \}$.

The *character* of the representation ϕ is a function $\chi : G \longrightarrow \mathbb{C}$ defined as $\chi_\phi(g) = Tr(\phi(g))$, where Tr

denotes the trace of a matrix. It follows from the definition of the character χ_ϕ that $\chi_\phi(gxg^{-1}) = \chi_\phi(x)$ for all $g, x \in G$, and it means that χ_ϕ is a *class function*, i.e., it is constant on conjugacy classes of G. A character of an irreducible representation is called an irreducible character.

The irreducible characters of a group, as their name suggests, characterize inequivalent irreducible representations of a group. The following orthogonality result is of fundamental importance in the representation theory of finite groups.

Theorem 3.1.1. *Let ϕ and ψ be irreducible representations of G. Then*

$$\langle \chi_\phi, \chi_\psi \rangle = \begin{cases} 1, & \text{if } \phi \sim \psi, \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

It follows that the irreducible characters of G form an orthonormal set of class functions. The number of inequivalent irreducible representations is the same as the number of inequivalent irreducible characters of the group G, which in turn is given by the number of distinct conjugacy classes of G.

Now onward *it is assumed that G is a finite abelian group*. Since G is assumed to be abelian, there are $|G|$ number of conjugacy classes of G, and so there are $|G|$ number of inequivalent irreducible characters of G. The set of these characters form a group of order $|G|$, known as the dual group of G. More precisely, the *dual group* of G, denoted by $\widehat{G}$, is defined as

$$\widehat{G} = \{\chi \mid \chi : G \longrightarrow S^1 \text{ is an irreducible character of } G\}. \tag{3.5}$$

The group operation is pointwise multiplication, i.e., $(\chi_1 \cdot \chi_2)(g) = \chi_1(g)\chi_2(g)$ for all $g \in G$ and $\chi_1, \chi_2 \in \widehat{G}$.

We have the following orthogonality relation.

Theorem 3.1.2. *Let $\chi_1, \chi_2 \in \widehat{G}$. Then*

$$\langle \chi_1, \chi_2 \rangle = \begin{cases} 1, & \text{if } \chi_1 = \chi_2, \\ 0, & \text{otherwise.} \end{cases} \tag{3.6}$$

The irreducible characters of G form a basis of $L(G)$ over $\mathbb{C}$, therefore any $f \in L(G)$ can be written as

$$f(g) = \sum_{\chi_i \in \widehat{G}} c_i \chi_i(g), \quad \text{where } c_i \in \mathbb{C} \text{ for all } \chi_i \in \widehat{G}. \tag{3.7}$$

It follows from Theorem 3.1.2 that $c_i = \langle f, \chi_i \rangle$. Therefore,

$$f(g) = \sum_{\chi_i \in \widehat{G}} \langle f, \chi_i \rangle \chi_i(g) = \frac{1}{\sqrt{|G|}} \sum_{\chi \in \widehat{G}} \widehat{f}(\chi)\chi(g), \tag{3.8}$$

where the Fourier transform $\widehat{f} : \widehat{G} \longrightarrow \mathbb{C}$ is defined below.

Definition 3.1.3. *Let G be a finite abelian group and $f \in L(G)$. Then the Fourier transform $\widehat{f} : \widehat{G} \longrightarrow \mathbb{C}$ is defined by*

$$\widehat{f}(\chi) = \sqrt{|G|}\langle f, \chi \rangle = \frac{1}{\sqrt{|G|}} \sum_{g \in G} f(g)\overline{\chi(g)}. \tag{3.9}$$

The Eq. (3.9) and Eq. (3.8) define the direct and inverse Fourier transform respectively. It can be verified that the map $f \longrightarrow \widehat{f}$ defines a linear transformation from $L(G)$ to $L(\widehat{G})$.

3.2 Walsh Functions as Characters of the Group $(\mathbb{Z}/2\mathbb{Z})^n$

Let $\omega_n = e^{\frac{2\pi i}{n}}$. Consider $G = \mathbb{Z}/n\mathbb{Z}$. Then $\widehat{G} = \{\chi_0, \chi_1, \ldots, \chi_{n-1}\}$, where $\chi_k(m) = \omega_n^{km}$ for $k = 0, 1, \ldots, n-1$. The well-known Discrete Fourier Transform is the Fourier transform for the complex valued functions defined on the group $G = \mathbb{Z}/n\mathbb{Z}$.

Example 3.2.1. *For $G = \mathbb{Z}/2\mathbb{Z} = \{[0], [1]\}$, the dual group $\widehat{G}$ consists of $\{\chi_0, \chi_1\}$ such that*

$$\chi_0([0]) = 1, \chi_0([1]) = 1, \quad \chi_1([0]) = 1, \chi_1([1]) = -1.$$

Here $[0]$ and $[1]$ denote the equivalence classes of 0 and 1, respectively. For easing the notation, in the following we will just write 0 and 1 for $[0]$ and $[1]$, respectively. The character table of the group $G\mathbb{Z}/2\mathbb{Z} = \{0, 1\}$ is shown in Table 1. We recall the Hadamard gate H, which sends the qubit $|0\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and the qubit $|1\rangle$ to

	0	1
χ_0	1	1
χ_1	1	-1

Table 1: The character table of $G = \mathbb{Z}/2\mathbb{Z} = \{0, 1\}$.

$\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. *It is interesting to note the similarity of the character table of $G = \mathbb{Z}/2\mathbb{Z}$ with the transformation matrix of the Hadamard gate H given by*

$$H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Let $\chi_i \in \widehat{G_1}$ and $\chi_j \in \widehat{G_2}$. Then $\chi_i \otimes \chi_j \in \widehat{G_1 \times G_2}$, such that $(\chi_i \otimes \chi_j)(g_1, g_2) = \chi_i(g_1)\chi_j(g_2)$. It can be shown that if $\widehat{G_1} = \{\chi_i \mid i = 1, \cdots m\}$ and $\widehat{G_2} = \{\chi_j \mid j = 1, \cdots n\}$, then $\widehat{G_1 \times G_2} = \{\chi_{ij} := \chi_i \otimes \chi_j \mid i = 1, \cdots m, j = 1, \cdots n\}$.

Example 3.2.2. *For $G = \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z} = \{0, 1\} \times \{0, 1\} = \{00, 01, 10, 11\}$, the dual group $\widehat{G} = \{\chi_{ij} = \chi_i \otimes \chi_j \mid i = 0, 1, j = 0, 1\}$. Again we note the similarity between the character table of $G = \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ (see Table 2) and the*

	00	01	10	11
χ_{00}	1	1	1	1
χ_{01}	1	-1	1	-1
χ_{10}	1	1	-1	-1
χ_{11}	1	-1	-1	1

Table 2: The character table of $G = \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z} = \{0, 1\} \times \{0, 1\} = \{00, 01, 10, 11\}$.

transformation matrix of

$$H \otimes H \equiv \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}.$$

The above examples can easily be generalized to the group $G = \{0, 1\}^n$, which is a group of order 2^n with elements identified with all the binary words of length n. The elements of this group can also be written as integers $0, 1, 2, \ldots, 2^n - 1$, where the integer k corresponds to the string $k_{n-1} k_{n-2} \cdots k_0$ which is the binary representation of the integer k. For example, for $n = 2$, the elements of $G = \{0, 1\}^2$ can be written as $\{0, 1, 2, 3\}$, where 0 is identified with 00, 1 is with 01, 2 is with 10 and 3 is with 11. A similar convention is followed for the character χ_k, i.e., $\chi_k = \chi_{k_{n-1}} \otimes \chi_{k_{n-2}} \otimes \cdots \otimes \chi_{k_1} \otimes \chi_{k_0} = \chi_{k_{n-1} k_{n-2} \ldots k_1 k_0}$. For example, χ_2 denotes $\chi_{10} = \chi_1 \otimes \chi_0$, and χ_3 denotes $\chi_{11} = \chi_1 \otimes \chi_1$. It is clear that using this convention for the group $G = (\mathbb{Z}/2\mathbb{Z})^n = \{0, 1\}^n$ the dual group is $\widehat{G} = \{\chi_k \mid k = 0, 1, \cdots 2^n - 1\}$. More explicitly, suppose k has the binary representation $k_{n-1} k_{n-2} k_1 k_0$ and $x \in G$ such that it has the binary representation $x_{n-1} x_{n-2} \cdots x_1 x_0$. Then

$$
\begin{aligned}
\chi_k(x) &= \left(\chi_{k_{n-1}} \otimes \chi_{k_{n-2}} \otimes \cdots \chi_{k_1} \otimes \chi_{k_0} \right)(x) \\
&= \chi_{k_{n-1}}(x_{n-1}) \chi_{k_{n-2}}(x_{n-2}) \cdots \chi_{k_1}(x_1) \chi_{k_0}(x_0) \\
&= (-1)^{\left(\sum_{i=0}^{n-1} x_i k_i \right)}
\end{aligned}
\tag{3.10}
$$

Definition 3.2.3. *Let $N = 2^n$ be a positive integer and let $G = (\mathbb{Z}/2\mathbb{Z})^n = \{0, 1\}^n$. Assume $0 \le k \le N$ to be an integer. Then the character $\chi_k \in \widehat{G}$ can be identified with the kth Walsh function of order N.*

Definition 3.2.4. *Let $N = 2^n$ be a positive integer. Suppose the interval $[0, 1]$ is divided into N equal sub-intervals each of length $\frac{1}{N}$. These sub-intervals are ordered from left to right by assigning binary equivalent of numbers from 0, 1, 2, ..., $N - 1$, with 0 assigned to the left most interval. It is clear that any $x \in [0, 1]$ will lie in one of these intervals. Then the kth Walsh function of order N is the function $W_k(x) : [0, 1] \longrightarrow \{1, -1\}$ defined by*

$$
W_k(x) := \left\{
\begin{array}{ll}
\chi_k(i), & \frac{i}{N} \le x < \frac{i+1}{N}, \, 0 \le i \le N - 1, \\
\chi_k(N - 1), & x = 1 .
\end{array}
\right.
\tag{3.11}
$$

Example 3.2.5. *For $N = 2^2$ there are 4 Walsh functions as shown in Figure 2. It is instructive to note the correspondence between the rows of the character table of $G = \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ in Table 2 and the graph of Walsh functions shown in Figure 2 below.*

We note that in Eq. (3.11) the kth Walsh function $W_k(x)$ of order N is defined in terms of characters $\chi_k \in \widehat{G}$, where $G = (\mathbb{Z}/2\mathbb{Z})^n = \{0, 1\}^n$ and $k = 0, 1, 2, \cdots N - 1$. It is clear from the above discussion that the character theory of finite groups provides valuable insights into the structure of Walsh functions.

3.3 Walsh-Hadamard Transform

Let $G = (\mathbb{Z}/2\mathbb{Z})^n = \{0, 1\}^n$ and $f \in L(G)$. Then from our earlier discussion (see Definition 3.1.3 and Eq. (3.10)) we have

$$
\widehat{f}(\chi_k) = \frac{1}{\sqrt{|G|}} \sum_{g \in G} f(g) \chi_k(g) = \frac{1}{\sqrt{N}} \sum_{m=0}^{2^n - 1} f(m) \chi_k(m) = \frac{1}{\sqrt{N}} \sum_{m=0}^{2^n - 1} f(m)(-1)^{\sum_{i=0}^{n-1} m_i k_i}.
\tag{3.12}
$$

where k has the binary representation $k_{n-1} k_{n-2} k_1 k_0$ and m has the binary representation $m_{n-1} m_{n-2} m_1 m_0$ and $N = |G| = 2^n$. Similarly, following Eq. (3.8), we have

$$
f(m) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \widehat{f}(\chi_k) \chi_k(m) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \widehat{f}(\chi_k)(-1)^{\sum_{i=0}^{n-1} m_i k_i}.
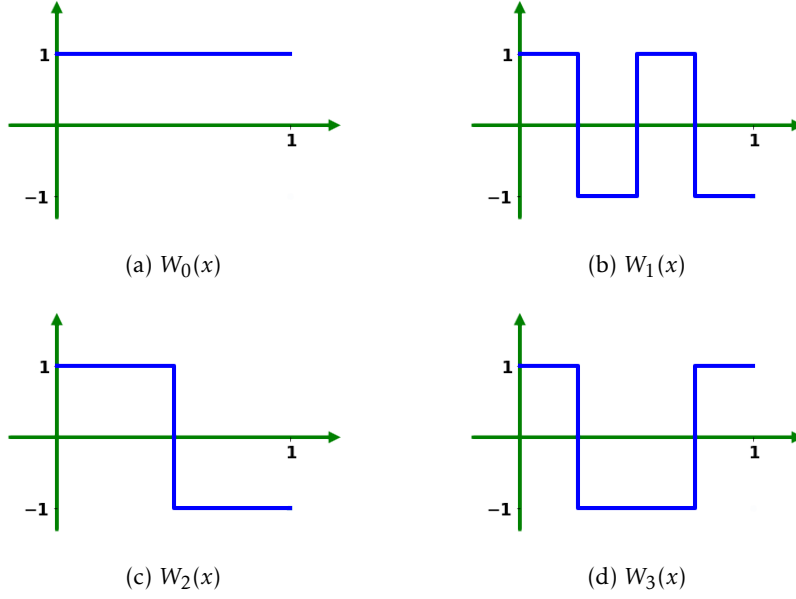\tag{3.13}
$$

(a) $W_0(x)$ (b) $W_1(x)$

(c) $W_2(x)$ (d) $W_3(x)$

Figure 2: Walsh functions in the natural (or Hadamard) ordering for $N = 4$.

(a) $W_0(x)$ (b) $W_1(x)$

(c) $W_2(x)$ (d) $W_3(x)$

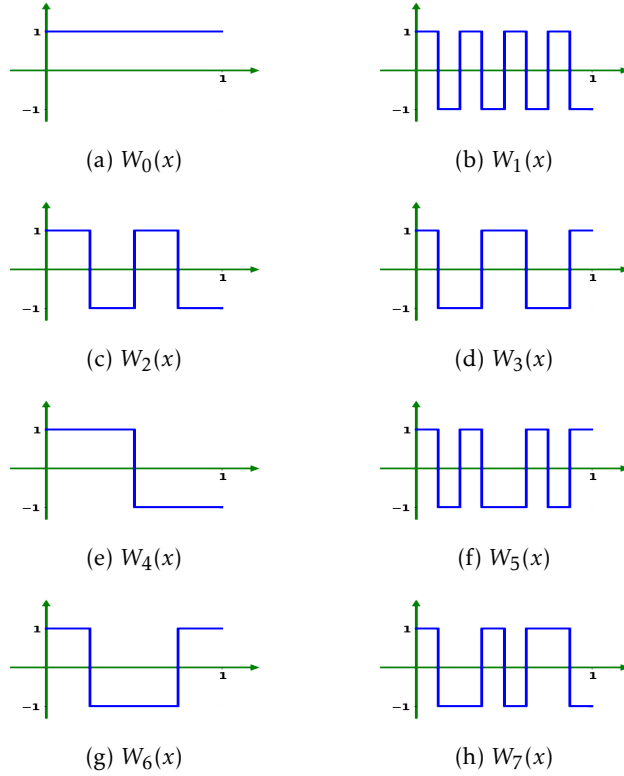(e) $W_4(x)$ (f) $W_5(x)$

(g) $W_6(x)$ (h) $W_7(x)$

Figure 3: Walsh functions in the natural (or Hadamard) ordering for $N = 8$.

Definition 3.3.1. *Let $\mathbf{v} = [f(0) \quad f(1) \quad f(2) \quad \cdots \quad f(N-1)]^T$ be a vector with $N = 2^n$ components. Then its Walsh-Hadamard transform is the vector defined by $\widehat{\mathbf{v}} = [\widehat{f}(\chi_0) \quad \widehat{f}(\chi_1) \quad \widehat{f}(\chi_2) \quad \cdots \quad \widehat{f}(\chi_{N-1})]^T$, where $\widehat{f}(\chi_k)$ is*

defined by Eq. (3.12), for $k = 0, 1, 2, \cdots N - 1$. Similarly, given $\widehat{\mathbf{v}}$, its inverse Walsh-Hadamard transform is $\mathbf{v}$ such that the component $f(m)$ of $\mathbf{v}$ is defined by Eq. (3.13).

We will write

$$[f(0) \quad f(1) \quad f(2) \quad \cdots \quad f(N-1)]^T \longleftrightarrow [\widehat{f}(\chi_0) \quad \widehat{f}(\chi_1) \quad \widehat{f}(\chi_2) \quad \cdots \quad \widehat{f}(\chi_{N-1})]^T$$

to denote the pair of a vector and its Walsh-Hadamard transform.

Example 3.3.2. *The Walsh-Hadamard transform of the vector $\mathbf{v} = [f(0) \quad f(1) \quad f(2) \quad f(3)]^T$ is*

$$\widehat{\mathbf{v}} = \frac{1}{2} \begin{pmatrix} f(0) + f(1) + f(2) + f(3) \\ f(0) - f(1) + f(2) - f(3) \\ f(0) + f(1) - f(2) - f(3) \\ f(0) - f(1) - f(2) + f(3) \end{pmatrix}.$$

One can compute this transform by computing

$$(H \otimes H)\, \mathbf{v} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{pmatrix}.$$

It can be checked that, Ex. 3.3.2 generalizes for computing the Walsh-Hadamard transform of an input vector of of size $N = 2^n$. In fact, for any integer $N = 2^n$, the Walsh-Hadamard transform of $\mathbf{v} = [f(0) \quad f(1) \quad \cdots \quad f(N-1)]^T$ is given by

$$(H^{\otimes n} \mathbf{v}). \tag{3.14}$$

A naive approach to compute the Walsh-Hadamard transform involving matrix-vector multiplication is of the order $\mathcal{O}(N^2)$ where $N = 2^n$. Although, in practice one computes the Walsh-Hadamard transform by employing a faster classical algorithm, namely the Fast Walsh-Hadamard Transform [11], [2]. The classical Fast Walsh-Hadamard Transform algorithm has the time complexity of the order of $\mathcal{O}(N \log_2(N))$ for computing the Walsh-Hadamard transform of an input vector of size $N = 2^n$.

3.4 Quantum Walsh-Hadamard Transform

The quantum implementation of Walsh-Hadamard transform involves two main steps. First, preparing the initial state $\sum_{k=0}^{N-1} f(k)|k\rangle$, and second applying Hadamard gates $H^{\otimes n}$ on it. Here $N = 2^n$ and it is also assumed that $\|f\| = 1$, or equivalently, $\sum_{k=0}^{N-1} (f(k))^2 = 1$. It can be verified that,

$$H^{\otimes n} \left[\sum_{k=0}^{N-1} f(k)|k\rangle \right] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \left(\sum_{m=0}^{N-1} f(m)(-1)^{\sum_{i=0}^{n-1} m_i k_i} \right) |k\rangle = \sum_{k=0}^{N-1} \widehat{f}(\chi_k) |k\rangle.$$

4 Hybrid Classical-Quantum Approach for Walsh-Hadamard Transform

A variation of quantum Walsh-Hadamard transform is at the heart of our algorithm to solve nonlinear differential and integral equations. Indeed, the Hadamard gate is one of the most useful quantum gates and the Walsh-Hadamard transform is the first step in many important quantum algorithms. It was discussed earlier that the Walsh-Hadamard transform of $\mathbf{v} = [f(0) \quad f(1) \quad \cdots \quad f(N-1)]^T$ is given by
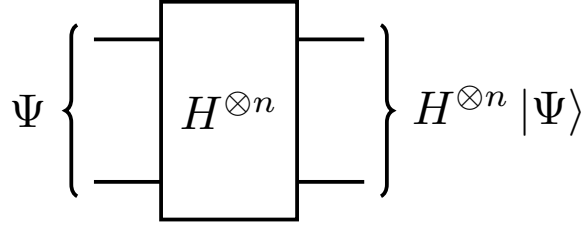
Figure 4: Quantum circuit for computing Walsh-Hadamard transform.

$(H^{\otimes n} \mathbf{v})$. Assuming that $\|\mathbf{v}\| = 1$, a simple quantum circuit consisting of n Hadamard gates can compute the Walsh-Hadamard transform of an input vector $\mathbf{v}$ of size $N = 2^n$ (see Fig. 4). However, the difficulty in this simple approach lies in the measurement. For example, consider the case of $n = 2$ qubits with $\mathbf{v} = [f(0) \quad f(1) \quad f(2) \quad f(3)]^T$. The circuit for computing the Walsh-Hadamard transform $\widehat{\mathbf{v}}$ in this case is shown in Figure 5. The input state for the circuit in Figure 5 is prepared to be

$$\mathbf{v} = f(0)|00\rangle + f(1)|01\rangle + f(2)|10\rangle + f(3)|11\rangle.$$
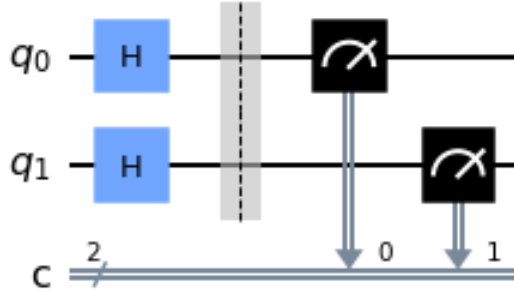

Figure 5: Quantum circuit for computing Walsh-Hadamard transform for $n = 2$ qubits.

However, one can only find the square of the amplitudes of the Walsh-Hadamard transform values by carrying out the measurement in this case. Since the input sequence is assumed to be real, the components of Walsh-Hadamard transformed vector $\widehat{\mathbf{v}}$ would also be real. However, the components of $\widehat{\mathbf{v}}$ may be positive or negative and this sign information is lost on carrying out the measurement. We tackle this problem by using the following simple observation.

Lemma 4.0.1. *Let*

$$[f(0) \quad f(1) \quad f(2) \quad \cdots \quad f(N-1)]^T \longleftrightarrow [\widehat{f}(\chi_0) \quad \widehat{f}(\chi_1) \quad \widehat{f}(\chi_2) \quad \cdots \quad \widehat{f}(\chi_{N-1})]^T$$

be a pair of a sequence and its Walsh-Hadamard transform. If

$$f(0) > \sum_{k=1}^{N-1} |f(k)|, \tag{4.1}$$

11

then all the components of $[\widehat{f}(\chi_0) \quad \widehat{f}(\chi_1) \quad \widehat{f}(\chi_2) \quad \cdots \quad \widehat{f}(\chi_{N-1})]^T$ *are positive.*

Proof. It is a consequence of the definition of Walsh-Hadamard transform that each component of the transformed vector

$$[\widehat{f}(\chi_0) \quad \widehat{f}(\chi_1) \quad \widehat{f}(\chi_2) \quad \cdots \quad \widehat{f}(\chi_{N-1})]^T$$

is of the form of a constant multiple of $f(0) \pm f(1) \pm f(2) \pm \quad \cdots \quad \pm f(N-1)$. The given hypothesis $f(0) > \sum_{k=1}^{N-1}|f(k)|$ then implies that all the components of the vector

$$[\widehat{f}(\chi_0) \quad \widehat{f}(\chi_1) \quad \widehat{f}(\chi_2) \quad \cdots \quad \widehat{f}(\chi_{N-1})]^T$$

are positive. □

Example 4.0.2. *The Walsh-Hadamard transform of* $\mathbf{v} = [f(0) \quad f(1) \quad f(2) \quad f(3)]^T$ *is*

$$\widehat{\mathbf{v}} = \frac{1}{2}\begin{pmatrix} f(0) + f(1) + f(2) + f(3) \\ f(0) - f(1) + f(2) - f(3) \\ f(0) + f(1) - f(2) - f(3) \\ f(0) - f(1) - f(2) + f(3) \end{pmatrix}.$$

It is clear that if $f(0) > |f(1)|+|f(2)|+|f(3)|$, *then all the components of the vector* $\widehat{\mathbf{v}}$ *are positive. In this case, one can directly compute the Walsh-Hadamard Transform by carrying out measurement in the circuit shown in Figure 5. Since all the components of the transformed vector are positive, there is no ambiguity about signs when computing the components from the probability measurements.*

From the above example, it follows that if the input vector $\mathbf{v} = [f(0) \quad f(1) \quad f(2) \quad \cdots \quad f(N-1)]^T$ is normalized (i.e., $\|\mathbf{v}\| = 1$) and Eq. (4.1) is satisfied, then it is easy to compute the Walsh-Hadamard transform. For example, the circuit shown in Figure 5 can be used for computing the Walsh-Hadamard transform for $N = 4$. Of course, the condition in Eq. (4.1) does not always hold. In such cases, we proceed as follows. Define $b_0 = \epsilon + \sum_{k=0}^{N-1}|f(k)|$, where ϵ is an arbitrary positive number. Then the Walsh-Hadamard transform of $\mathbf{v_1} = [b_0 \quad f(1) \quad f(2) \quad \cdots \quad f(N-1)]^T$ is given by

$$\widehat{\mathbf{v_1}} = \widehat{\mathbf{v}} + \widehat{\mathbf{t}}, \qquad \text{where } \widehat{\mathbf{t}} = \delta[1 \quad 1 \quad \cdots \quad 1]^T \text{ is vector of size } N \text{ with } \delta = \frac{1}{\sqrt{N}}(b_0 - f(0)). \qquad (4.2)$$

We note that vectors $\mathbf{v}$ and $\mathbf{v_1}$ differ only in the first component. A quantum state is prepared based on the normalized state vector $\frac{\mathbf{v_1}}{\|\mathbf{v_1}\|}$. Clearly, all components of the normalized state vector $\frac{\mathbf{v_1}}{\|\mathbf{v_1}\|}$ are non-negative and the condition given in Eq. (4.1) is satisfied. Therefore, the Walsh-Hadamard transform $\frac{\widehat{\mathbf{v_1}}}{\|\mathbf{v_1}\|}$ can be obtained using the quantum Hadamard gates and measurement operations without any ambiguity about the signs. Then using Eq. (4.2), one can compute $\widehat{\mathbf{v}} = \widehat{\mathbf{v_1}} - \widehat{\mathbf{t}}$. The hybrid classical-quantum approach explained above is employed in Algorithm 1 to compute the Walsh-Hadamard transform of a given input

vector.

Algorithm 1: A hybrid classical-quantum algorithm for computing the Walsh-Hadamard transform $\mathcal{H}_Q(A)$ of a given input vector A.

Input: The input vector $A = [a_0 \quad a_1 \quad a_2 \quad \cdots \quad a_{N-1}]^T$ where $N = 2^n$ is a positive integer and $a_i \in \mathbb{R}$ for $i = 0$ to $i = N - 1$.

Output: The Walsh-Hadamard transform of the input vector.

1 **Function** $\underline{\mathcal{H}_Q\,(A)}$

2     $b_0 = \epsilon + \sum_{k=0}^{N-1} |a_k|$ `// Here ε is any positive number.`

3     $c = \sqrt{\left[b_0^2 + \sum_{k=1}^{N-1} a_k^2 \right]}$ `// Let` $\widetilde{A} = [b_0 \quad a_1 \quad a_2, \cdots \quad a_{N-1}]^T$`. Then` $c = \|\widetilde{A}\|$

4   Prepare the state $|\Psi\rangle = \frac{b_0}{c}|0\rangle + \sum_{k=1}^{N-1} \frac{a_k}{c}|k\rangle$ using n qubits.   `// Initialize the state` $|\Psi\rangle$ `with` $\frac{\widetilde{A}}{\|\widetilde{A}\|}$`.`

5   Apply $H^\otimes$ on $|\Psi\rangle$.

6   Measure all the n qubits to compute the probability p_k of obtaining the state $|k\rangle$, for $k = 0$ to $2^n - 1$.

7     $\delta = \frac{1}{\sqrt{N}}(b_0 - a_0)$

8   **return** $\underline{\text{the vector } [c\sqrt{p_0} - \delta \quad c\sqrt{p_1} - \delta \quad c\sqrt{p_2} - \delta \quad \cdots \quad c\sqrt{p_{N-1}} - \delta]^T}$

We note that the parameter ϵ ensures that Algorithm 1 also works for the special case when the $\|A\| = 0$.

4.1 Computational Complexity

The computational complexity of the classical Fast Walsh-Hadamard Transform [11] for an input vector of size N is of the order of $\mathcal{O}(N \log_2(N))$ additions and subtractions. On the other hand, in our proposed hybrid classical-quantum algorithm the Walsh-Hadamard transform for an input vector of size N involves computations of $\mathcal{O}(N)$ additions, multiplications and square roots. In order to make a more precise comparison of the performances of the classical and the hybrid algorithms, consider an input vector of size N such that each of the N components is K bits. Since the addition or subtraction of two K bits numbers is of the order of $\mathcal{O}(K)$ using the elementary school algorithms of addition and subtraction with carry and borrow, respectively, the classical Fast Walsh-Hadamard Transform will be of the order of $\mathcal{O}(N \log_2(N)K)$. Next we note that the complexity of computing the square-root of a number is of the same order as that of multiplication using Newton's method. Since multiplication of two K bits number if of the order of $K \log_2(K)$ as given by Harvey-Hoeven algorithm [15], our proposed hybrid classical-quantum algorithm, Algorithm 1, is of the order of $\mathcal{O}(NK \log_2(K))$. It follows that for a fixed K our proposed algorithm is of the order of $\mathcal{O}(N)$, whereas as mentioned earlier, the classical Fast Walsh-Hadamard Transform is of the order of $\mathcal{O}(N \log_2(N))$.

Remark 4.1.1.

(a) *The Harvey-Hoeven algorithm for multiplication [15] is of order $O(K \log_2(K))$ only for extremely large numbers. The algorithm is $O(K \log_2(k))$ only if $K \geq 2^{d^{12}}$ with $d = 1729$. Although they describe modifications in their algorithm, which will reduce d to 9.*

(b) *Even if one uses the classical Karatsuba algorithm, which is of the order of $\mathcal{O}(K^{1.585})$, in computing multiplications and square-roots in Algorithm 1, it is obvious that Algorithm 1 will be of $\mathcal{O}(N)$.*

In the next section, using some simple examples, we illustrate how Walsh-Hadamard basis functions can be used for representing functions and performing integration and differentiation operations on them.

This will set the stage for our proposed hybrid classical-quantum approach for the solution of nonlinear ordinary differential equations using Walsh-Hadamard basis functions and Algorithm 1.

5 Walsh-Hadamard Basis: Representation, Integration and Differentiation of arbitrary functions

Any piecewise continuous function f on $[0,1]$ can be approximated by a discretizing the function and then the discrtized version can be be expressed as a linear combination of Walsh functions. To demonstrate the use of Walsh-Hadamard transform for representation and integration of functions, especially in the context of the proposed hybrid classical-quantum approaches, we provide some illustrative examples. It is important to note that we consider the natural ordering of Walsh functions in our formulations unlike most works in the literature that are based on the sequency ordering. This consideration results in expressions for integration and differentiation matrices (as discussed below) that are slightly different from those commonly available in the literature (e.g., see [3], [1], [12], [13]).

Example 5.0.1. *Consider the function* $\cos(\pi t)$ *on* $0 \le t \le 1$. *The interval* $[0,1]$ *is divided into* $N = 4$ *sub-intervals and the function* $\cos(\pi t)$ *is discretized as*

$$\cos(\pi t) \simeq \left[\cos\left(\frac{\pi}{8}\right) \quad \cos\left(\frac{3\pi}{8}\right) \quad \cos\left(\frac{5\pi}{8}\right) \quad \cos\left(\frac{7\pi}{8}\right)\right].$$

Then the computation of Walsh-Hadamard transform gives

$$\left[\cos\left(\frac{\pi}{8}\right) \quad \cos\left(\frac{3\pi}{8}\right) \quad \cos\left(\frac{5\pi}{8}\right) \quad \cos\left(\frac{7\pi}{8}\right)\right] \longleftrightarrow \frac{1}{2}[0 \quad 1.08 \quad 2.61 \quad 0].$$

It follows from the above calculation and and Eq. (3.8) *(with* $|G| = N = 4$*) that*

$$\cos(\pi t) \simeq \frac{1}{4}\left(1.08\,W_1(t) + 2.61\,W_2(t)\right).$$

Of course, increasing the value of N improves the approximations obtained using the Walsh functions. Figure 6 below shows that graph of function $\cos(\pi t)$ *for* $0 \le t \le 1$ *and the approximations of this function obtained by using Walsh functions of order N = 2, 4, 8 and 16.*

5.1 Integration of Walsh functions

We have seen that Walsh functions can be used to approximate the discretized version of any continuous function defined on $[0,1]$. One can use Walsh functions to approximate the integrals of piecewise continuous functions on $[0,1]$. For example, consider the function $f(t) = \int_0^t W_0(x)\,dx = t$, with $0 \le t \le 1$. It can be discretized over $[0,1]$, using $N = 4$ as

$$f(t) = t \simeq \left[\frac{1}{8} \quad \frac{3}{8} \quad \frac{5}{8} \quad \frac{7}{8}\right]^T,$$

and the Walsh-Hadamard transform computation gives

$$t \simeq \left[\frac{1}{8} \quad \frac{3}{8} \quad \frac{5}{8} \quad \frac{7}{8}\right]^T \longleftrightarrow \frac{1}{2}\left[2 \quad -\frac{1}{2} \quad -1 \quad 0\right]^T.$$
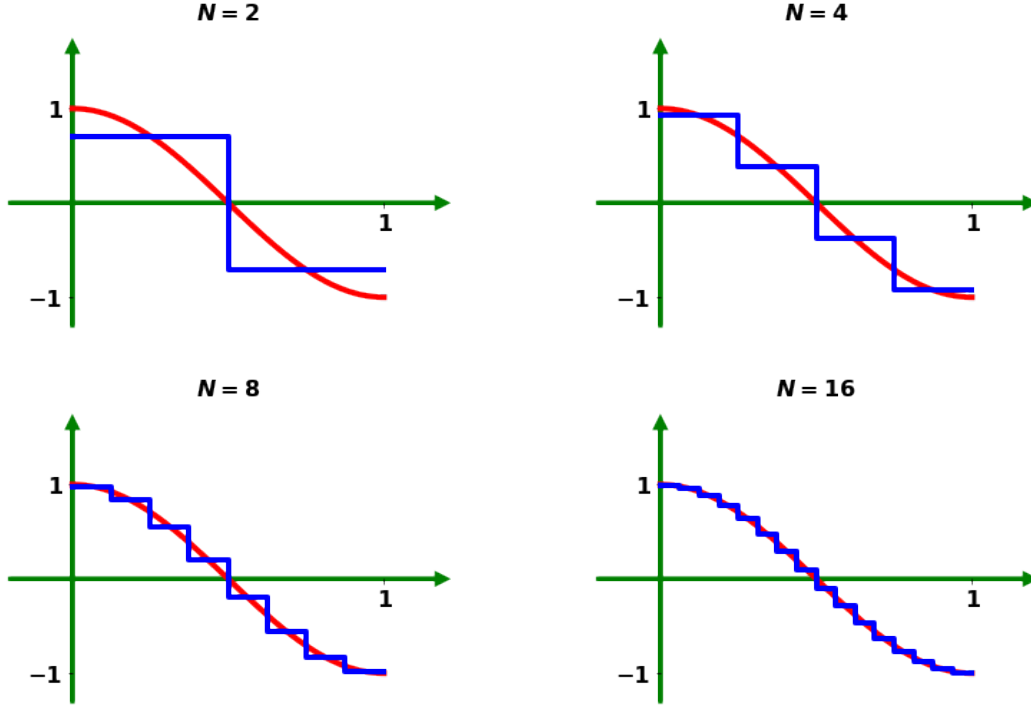
14

Figure 6: Approximation of $f(t) = \cos(\pi t)$, for $0 \le t \le 1$, using Walsh functions of order $N = 2, 4, 8$ and 16. The function $\cos(\pi t)$ is plotted in red and the Walsh function based approximations are plotted in blue.

From the above computation and Eq. (3.8) (with $|G| = N = 4$) it follows that

$$\int_0^t W_0(x)\,dx = t = \frac{1}{4}\left(2W_0(t) - \frac{1}{2}W_1(t) - W_2(t)\right) = \frac{1}{2}W_0(t) - \frac{1}{8}W_1(t) - \frac{1}{4}W_2(t). \tag{5.1}$$

Similarly, it can be checked that

$$\int_0^t W_1(x)\,dx \simeq \begin{bmatrix}\frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8}\end{bmatrix}^T \longleftrightarrow \frac{1}{2}\begin{bmatrix}\frac{1}{2} & 0 & 0 & 0\end{bmatrix}^T.$$

This gives,

$$\int_0^t W_1(x)\,dx = \frac{1}{8}W_0(t). \tag{5.2}$$

We have

$$\int_0^t W_2(x)\,dx \simeq \begin{bmatrix}\frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8}\end{bmatrix}^T \longleftrightarrow \frac{1}{2}\begin{bmatrix}1 & 0 & 0 & -\frac{1}{2}\end{bmatrix}^T,$$

and

$$\int_0^t W_3(x)\,dx \simeq \begin{bmatrix}\frac{1}{8} & \frac{1}{8} & -\frac{1}{8} & -\frac{1}{8}\end{bmatrix}^T \longleftrightarrow \frac{1}{2}\begin{bmatrix}0 & 0 & \frac{1}{2} & 0\end{bmatrix},$$

and similar calculations give,

$$\int_0^t W_2(x)\,dx = \frac{1}{4}W_0(t) - \frac{1}{8}W_3(t), \tag{5.3}$$

$$\int_0^t W_3(x)\,dx = \frac{1}{8}W_2(t). \tag{5.4}$$

The above calculations give rise to the following *integration matrix*

$$I_4 = \begin{pmatrix} \frac{1}{2} & \frac{1}{8} & \frac{1}{4} & 0 \\ -\frac{1}{8} & 0 & 0 & 0 \\ -\frac{1}{4} & 0 & 0 & \frac{1}{8} \\ 0 & 0 & -\frac{1}{8} & 0 \end{pmatrix}. \tag{5.5}$$

The function to be integrated can be represented as a column vector with the Walsh functions as the basis functions. Then the multiplying this column vector by the integration matrix is equivalent to performing the integration of the function.

Example 5.1.1. *The function* $\cos t$ *can be discretized and represented as* $\left[\cos\left(\frac{1}{8}\right) \quad \cos\left(\frac{3}{8}\right) \quad \cos\left(\frac{5}{8}\right) \quad \cos\left(\frac{7}{8}\right)\right]$ *on* $[0,1]$ *for* $N = 4$*. Performing the Walsh-Hadamard transform gives*

$$\left[\cos\left(\frac{1}{8}\right) \quad \cos\left(\frac{3}{8}\right) \quad \cos\left(\frac{5}{8}\right) \quad \cos\left(\frac{7}{8}\right)\right] \longleftrightarrow \frac{1}{2}[3.375 \quad 0.232 \quad 0.471 \quad -0.108].$$

This means

$$\cos t \simeq \frac{1}{4}\left(3.375\,W_0(t) + 0.232\,W_1(t) + 0.471\,W_2(t) - 0.108\,W_3(t)\right).$$

Therefore the column coordinate vector for $\cos t$ *in terms of Walsh basis vectors is given by*

$$\mathbf{v} = [0.844 \quad 0.058 \quad 0.118 \quad -0.027]^T,$$

and

$$I_4\mathbf{v} = \begin{pmatrix} \frac{1}{2} & \frac{1}{8} & \frac{1}{4} & 0 \\ -\frac{1}{8} & 0 & 0 & 0 \\ -\frac{1}{4} & 0 & 0 & \frac{1}{8} \\ 0 & 0 & -\frac{1}{8} & 0 \end{pmatrix} \begin{pmatrix} 0.844 \\ 0.058 \\ 0.118 \\ -0.027 \end{pmatrix} = \begin{pmatrix} 0.459 \\ -0.105 \\ -0.214 \\ -0.015 \end{pmatrix}.$$

Therefore, we have

$$\int_0^t \cos x\,dx \simeq 0.459\,W_0(t) - 0.105\,W_1(t) - 0.214\,W_2(t) - 0.015\,W_3(t).$$

One can transform this back to obtain the discretization of $\int_0^t \cos x\,dx = \sin t$ *in the time domain as follows.*

$$2(H \otimes H)\begin{pmatrix} 0.459 \\ -0.105 \\ -0.214 \\ -0.015 \end{pmatrix} = \begin{pmatrix} 0.125 \\ 0.366 \\ 0.585 \\ 0.767 \end{pmatrix} = \begin{pmatrix} \sin(\frac{1}{8}) \\ \sin(\frac{3}{8}) \\ \sin(\frac{5}{8}) \\ \sin(\frac{7}{8}) \end{pmatrix}.$$

16

A similar calculation shows that for $N = 8$ the integration matrix I_8 is given by

$$I_8 = \begin{pmatrix} \frac{1}{2} & \frac{1}{16} & \frac{1}{8} & 0 & \frac{1}{4} & 0 & 0 & 0 \\ -\frac{1}{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{8} & 0 & 0 & \frac{1}{16} & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{16} & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{4} & 0 & 0 & 0 & 0 & \frac{1}{16} & \frac{1}{8} & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{16} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{8} & 0 & 0 & \frac{1}{16} \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{16} & 0 \end{pmatrix}.$$

The integration matrix I_N for $N = 2^n$ can be be computed similarly. The integration matrix I_N is a very sparse matrix and therefore computation of integration is not very costly even when n is large. It is easy to see that if $f(t)$ is discretized as $\mathbf{f} = [f\left(\frac{1}{2N}\right) \quad f\left(\frac{3}{2N}\right) \quad f\left(\frac{5}{2N}\right) \quad \cdots \quad f\left(\frac{2N-1}{2N}\right)]^T$, then $\int_0^t f(x)\,dx$ in the discretized form can be computed as

$$H^{\otimes n}(I_N H^{\otimes n}(\mathbf{f})), \tag{5.6}$$

for $0 \le t \le 1$. In our proposed approach based on a hybrid classical-quantum implementation of the Walsh-Hadamard transform as outlined in Algorithm 1, the above expression can be evaluated as

$$\mathcal{H}_Q(I_N \mathcal{H}_Q(\mathbf{f})). \tag{5.7}$$

Although we have provided examples in the domain $[0, 1]$, the present approach can easily be rescaled for arbitrary domains $[t_{\text{initial}}, t_{\text{final}}]$.

5.2 Differentiation Matrix

Since differentiation can be viewed as an inverse process of integration, the *differentiation matrix* can be obtain by finding the inverse of the integration matrix. For example, for $N = 4$, the differentiation matrix D_4 is given by

$$D_4 = I_4^{-1} = \begin{pmatrix} 0 & -8 & 0 & 0 \\ 8 & 32 & 0 & 16 \\ 0 & 0 & 0 & -8 \\ 0 & -16 & 8 & 0 \end{pmatrix}.$$

Similarly, the differentiation matrix D_N of order $N = 2^n$ can be be computed as the inverse of I_N. The differentiation matrix D_N is also a sparse matrix and therefore computation of differentiation is also not very costly even when n is large.

6 A Hybrid Algorithm for Solving Differential Equations

Walsh functions and the Walsh-Hadamard transform can be used to solve nonlinear differential equations. These methods are well-known and were used by several authors for solving differential equa-

tions [12], [13], [3]. The key idea of our work is to replace the classical computation of Walsh-Hadamard transform with a hybrid classical-quantum computation of Walsh-Hadamard transform (based on the natural ordering). Since the hybrid classical-quantum version of Walsh-Hadamard transform can be carried out more efficiently (in ($\mathcal{O}(N)$) operations) than the classical Fast Walsh-Hadamard transform ($\mathcal{O}(N\log_2(N))$), it provides a superior algorithm to solve nonlinear differential equations.

Consider the following system of nonlinear ordinary differential equations

$$\frac{dx_i}{dt} = f_i(x_1, x_2, \cdots, x_m, t), \quad \text{for } i = 1, 2, \cdots, m, \tag{6.1}$$

with the initial conditions

$$x_i(0) = q_i. \tag{6.2}$$

Algorithm 2: A hybrid classical-quantum algorithm for solving the system of differential equation given in Eq. (6.1).

Input: Integers N and N_{max}.

Output: Discretized versions of $x_i(t)$, with N sub-intervals on $0 \le t \le 1$, for $i = 1, 2, \cdots, m$.

/* The algorithm uses the quantum subroutine $\mathcal{H}_Q(\mathbf{v})$ to compute the quantum Walsh-Hadamard transform of the input vector $\mathbf{v}$. */

1 $\quad t = \begin{bmatrix} \frac{1}{2N} & \frac{3}{2N} & \frac{5}{2N} & \cdots & \frac{2N-1}{N} \end{bmatrix}^T$ // Initialize t.

2 **for** $i \leftarrow 1$ **to** m **do**

3 $\quad \Big\lfloor \quad x_i = x_{i\,\text{Initial}} = \begin{bmatrix} q_i & q_i & q_i \cdots & q_i \end{bmatrix}^T$ // Initialize x_i.

4 $\quad I_N =$ The integration matrix of order N // Initialize the integration matrix I_N.

5 **for** $i \leftarrow 1$ **to** N_{max} **do**

6 $\quad \Big\lfloor \quad x_i = x_{i\,\text{Initial}} + \mathcal{H}_Q\Big(I_N\, \mathcal{H}_Q(f_i(x_1, x_2, \cdots, x_m, t))\Big)$ // Compute $x_i = x_{i\,\text{Initial}} + \int_0^t f_i\, d\tau$.

7 **return** $\underline{x_1, x_2, \cdots, x_m.}$

Our algorithm starts by initializing x_i to a vector $x_{i\,\text{Initial}} = \begin{bmatrix} q_i & q_i & q_i & \cdots & q_i \end{bmatrix}^T$ of size N for $i = 1$ to $i = m$. Then using Eq. (6.1) and Eq. (6.2) one can write

$$x_i = x_{i\,\text{Initial}} + \int_0^t f_i(x_1, x_2, \cdots, x_m, \tau)\, d\tau$$

The Walsh functions based representation and the integration matrix approach, discussed earlier in Section 5, can be used to solve the above equations iteratively. The iterations can then be continued up to a maximum chosen number of times, N_{max}, where N_{max} depends on the accuracy of the solution desired. One can select N_{max} such that the successive iterations give identical results to the required number of decimal places. We note that the statement,

$$x_i = x_{i\,\text{Initial}} + \mathcal{H}_Q\Big(I_N\, \mathcal{H}_Q(f_i(x_1, x_2, \cdots, x_m, t))\Big)$$

in Algorithm 2, computes $x_i = x_{i\,\text{Initial}} + \int_0^t f_i(x_1, x_2, \cdots, x_m, t)\, d\tau$ (see Eq. (5.7) and the discussion before that). Here, we also note that $f_i(x_1, x_2, \cdots, x_m, t)$ is computed point-wise in the time-domain.

7 Computational Examples

In this section, we will give computational examples to illustrate Algorithm 2 for solving nonlinear differential equations. The proposed algorithms for the solution of nonlinear ordinary differential equations were successfully implemented and tested using the simulated environment of Qiskit (IBM's open source quantum computing platform).

7.1 Example 1 - Riccati differential equation

We consider the following Riccati differential equation for $0 \leq t \leq 1$,

$$\frac{dx_1}{dt} = x_1^2 + x_1 + 1, \tag{7.1}$$

with the initial conditions $x_1(0) = -\frac{1}{2}$. The analytical solution for the above initial value problem is

$$x_1 = \frac{1}{2}\left(\sqrt{3}\tan\left(\frac{\sqrt{3}t}{2}\right) - 1\right).$$

Our algorithm starts by initializing $x_1 = x_{1\,\text{Initial}} = [-\frac{1}{2} \quad -\frac{1}{2} \quad -\frac{1}{2} \quad -\frac{1}{2}]^T$. Then using Eq. (7.1) one can write

$$x_1(t) = x_{1\,\text{Initial}} + \int_0^t f_1(x_1(\tau))\, d\tau, \tag{7.2}$$

where $f_1(x_1(\tau)) = x_1(\tau)^2 + x_1(\tau) + 1$ The last expression can now can be used iteratively, employing the Walsh functions based representation and the integration matrix approach as discussed earlier in Section 5.1. The iterations can then be continued until the desired accuracy is achieved. The maximum number of iteration N_{max} can be chosen so that the successive iterations produce identical results up to the desired number of decimal places. The value N_{max} is chosen to be 10 in Algorithm 3.

The statement $x_1 = x_{1\,\text{Initial}} + \mathcal{H}_Q\left(I_4\,\mathcal{H}_Q(x_1^2 + x_1 + 1)\right)$ in Algorithm 3 computes $x_1(t)$ by evaluating the right side of Eq. (7.2). After the first iteration the result obtained is

$$x_1 = [-0.40625 \quad -0.21875 \quad -0.03125 \quad 0.15625].$$

The result obtained after the 10-th iteration, and rounded to the five decimal places, is

$$x_1 = [-0.40512 \quad -0.20567 \quad 0.02743 \quad 0.33735].$$

The correct analytic solution $x_1(t)$ along with solutions obtained by Walsh functions of order $N = 2, 4, 8$ and $N = 16$ are shown in Figure 7.
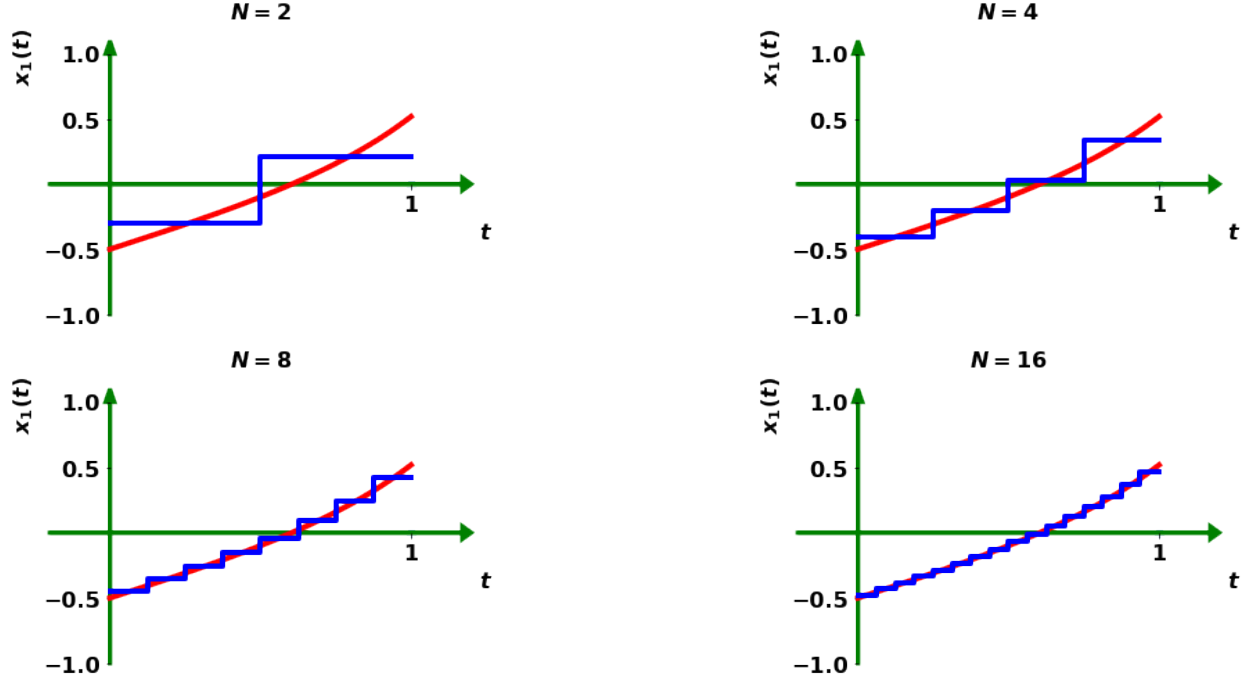
Figure 7: Analytic solution for $x_1(t)$ (in red) and the solutions obtained by Walsh functions of order $N = 2$, 4, 8 and $N = 16$ (in blue), based on the proposed hybrid classical-quantum approach.

Algorithm 3: A hybrid algorithm for solving the differential equation given in Eq. (7.1).

Input: $N = 4$ and $N_{max} = 10$.

Output: Discretized versions of $x_1(t)$ on $0 \le t \le 1$ with $N = 4$ sub-intervals.

```
/* The algorithm uses the quantum subroutine H_Q(v) to compute the quantum Walsh-Hadamard transform of
   the input vector v.                                                                                */
```

1   $t = \begin{bmatrix} \frac{1}{8} & \frac{3}{8} & \frac{5}{8} & \frac{7}{8} \end{bmatrix}^T$ `// Initialize t.`

2   $x_1 = x_{1\,\text{Initial}} = \begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix}^T$ `// Initialize x_1.`

3   $I_4 = \begin{pmatrix} \frac{1}{2} & \frac{1}{8} & \frac{1}{4} & 0 \\ -\frac{1}{8} & 0 & 0 & 0 \\ -\frac{1}{4} & 0 & 0 & \frac{1}{8} \\ 0 & 0 & -\frac{1}{8} & 0 \end{pmatrix}$ `// Initialize the integration matrix I_4.`

4   **for** $i \leftarrow 1$ **to** N_{max} **do**

     `/* Compute` $x_1 = x_{1\,\text{Initial}} + \int_0^t x_1(\tau)^2 + x_1(\tau) + 1\, d\tau.$ `*/`

5      $x_1 = x_{1\,\text{Initial}} + \mathcal{H}_Q\big(I_4\, \mathcal{H}_Q(f_1)\big)$, where $f_1(x_1) = x_1^2 + x_1 + 1$

6   **return** x_1.

7.2   Example 2 - A system of nonlinear ODEs

In this section, we will demonstrate the application of our proposed hybrid classical-quantum approach to the solution of a system of nonlinear ordinary differential equations involving two dependent variables.

Consider the following system of differential equations [3]

$$\frac{dx_1}{dt} = x_2, \tag{7.3}$$

$$\frac{dx_2}{dt} = -(3x_1x_2 + x_1^3), \tag{7.4}$$

with the initial conditions $x_1(0) = 0$ and $x_2(0) = 1$.

Algorithm 4: A hybrid algorithm for solving the system of differential equation given in Eq. (7.3) and Eq. (7.4).

Input: $N = 4$ and $N_{max} = 20$.

Output: Discretized versions of $x_1(t)$ and $x_2(t)$ on $0 \le t \le 1$ with $N = 4$ sub-intervals.

/* The algorithm uses the quantum subroutine $\mathcal{H}_Q(\mathbf{v})$ to compute the quantum Walsh–Hadamard transform of the input vector $\mathbf{v}$. */

1   $t = \begin{bmatrix} \frac{1}{8} & \frac{3}{8} & \frac{5}{8} & \frac{7}{8} \end{bmatrix}^T$ // Initialize t.

2   $x_1 = x_{1\,\text{Initial}} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$ // Initialize x_1.

3   $x_2 = x_{2\,\text{Initial}} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$ // Initialize x_2.

4   $I_4 = \begin{pmatrix} \frac{1}{2} & \frac{1}{8} & \frac{1}{4} & 0 \\ -\frac{1}{8} & 0 & 0 & 0 \\ -\frac{1}{4} & 0 & 0 & \frac{1}{8} \\ 0 & 0 & -\frac{1}{8} & 0 \end{pmatrix}$ // Initialize the integration matrix I_4.

5   **for** $i \leftarrow 1$ **to** N_{max} **do**

   /* Compute $x_1 = x_{1\,\text{Initial}} + \int_0^t x_2(\tau)d\tau$. */

6    $x_1 = x_{1\,\text{Initial}} + \mathcal{H}_Q\big(I_4\,\mathcal{H}_Q(f_1)\big)$, where $f_1(x_1, x_2) = x_2$

   /* Compute $x_2 = x_{2\,\text{Initial}} - \int_0^t \big(3x_1(\tau)x_2(\tau) + x_1(\tau)^3\big)\,d\tau$. */

7    $x_2 = x_{2\,\text{Initial}} + \mathcal{H}_Q\big(I_4\,\mathcal{H}_Q(f_2)\big)$, where $f_2(x_1, x_2) = -(3x_1x_2 + x_1^3)$

8   **return** x_1, x_2

Our algorithm begins by initializing $x_1 = x_{1\,\text{Initial}} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$ and $x_2 = W_0(t) = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$. Then using Eq. (7.3) and Eq. (7.4) one can write

$$x_1 = x_{1\,\text{Initial}} + \int_0^t f_1(x_1(\tau), x_2(\tau))\,d\tau,$$

$$x_2 = x_{2\,\text{Initial}} + \int_0^t f_2(x_1(\tau), x_2(\tau))\,d\tau,$$

where $f_1(x_1(\tau), x_2(\tau)) = x_2(\tau)$ and $f_2(x_1(\tau), x_2(\tau)) = -(3x_1(\tau)x_2(\tau) + x_1(\tau)^3)$. The last two equations can be solved using the Walsh functions based representation and the integration matrix approach, as discussed earlier in Section 5.1. The iterations can then be continued up to a maximum chosen number of times, N_{max}, which depends on the accuracy of the solution desired. For example, N_{max} can be chosen such that the successive iterations give identical results to the required number of decimal places. At the end of the first iteration we obtain

$$x_1 \simeq \begin{bmatrix} 0.125 & 0.375 & 0.625 & 0.875 \end{bmatrix}^T,$$

$$x_2 \simeq \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T.$$

We note that in Algorithm 4 the integration $x_1 = x_{1\text{Initial}} + \int_0^t f_1(x_1(\tau), x_2(\tau)) \, d\tau$ is performed by the statement

$$x_1 = x_{1\text{Initial}} + \mathcal{H}_Q\left(I_4 \, \mathcal{H}_Q(f_1)\right).$$

Similarly, the statement

$$x_2 = x_{2\text{Initial}} + \mathcal{H}_Q\left(I_4 \, \mathcal{H}_Q(f_2)\right)$$

in Algorithm 4 is used to compute $x_2 = x_{2\text{Initial}} - \int_0^t f_2(x_1(\tau), x_2(\tau)) \, d\tau$. Here, we also note that computation $f_2(x_1, x_2) = -(3x_1 x_2 + x_1^3)$ is carried out point-wise in the time-domain. For example, when

$$x_1 = [0.125, 0.375, 0.625, 0.875]^T \text{ and } x_2 = [1, 1, 1, 1]^T,$$

we have

$$x_1^3 = [0.001953125, 0.052734375, 0.244140625, 0.669921875]^T, 3x_1 x_2 = [0.375, 1.125, 1.875, 2.625]^T,$$

and

$$-(3x_1 x_2 + x_1^3) = [-0.376953125, -1.177734375, -2.119140625, -3.294921875]^T.$$

After 8-th iteration the result is

$$x_1 = [0.11960814, 0.33997528, 0.51224524, 0.62590886],$$
$$x_2 = [0.95686836, 0.80607053, 0.57178512, 0.33552362],$$

which is the same as obtained in [3]. We note that the result after the 20-th iteration is

$$x_1 = [0.11960845, 0.33997421, 0.51220193, 0.62564211],$$
$$x_2 = [0.95686757, 0.80605858, 0.57176313, 0.33575831].$$

It is known that in this case the analytic solutions are $x_1(t) = \frac{2t}{t^2+2}$ and $x_2(t) = \frac{4-2t^2}{(t^2+2)^2}$. The analytic solutions $x_1(t)$ and $x_2(t)$ along with solutions obtained by Walsh functions of order $N = 4$ and $N = 16$ are shown in Figure 8.

8   Conclusion

In this work, we proposed a hybrid classical-quantum approach for solutions of nonlinear ordinary differential equations using Walsh-Hadamard basis functions. The Walsh-Hadamard transform is a key step in many known classical methods for the solution of nonlinear ordinary differential equations using the Walsh-Hadamard basis functions. While Hadamard gates are commonly used in many quantum algorithms and can naturally compute the Walsh-Hadamard transform (under certain conditions), there are challenges involved in extracting useful classical information. These challenges associated with obtaining Walsh-Hadamard transforms of arbitrary vectors are overcome via our proposed hybrid classical-quantum approach (see Algorithm 1). This hybrid classical-quantum approach for Walsh-Hadamard transform involves shifting, scaling and measurement operations (along with state preparation and use of quantum Hadamard gates). This hybrid classical-quantum approach for obtaining the Walsh-Hadamard trans-
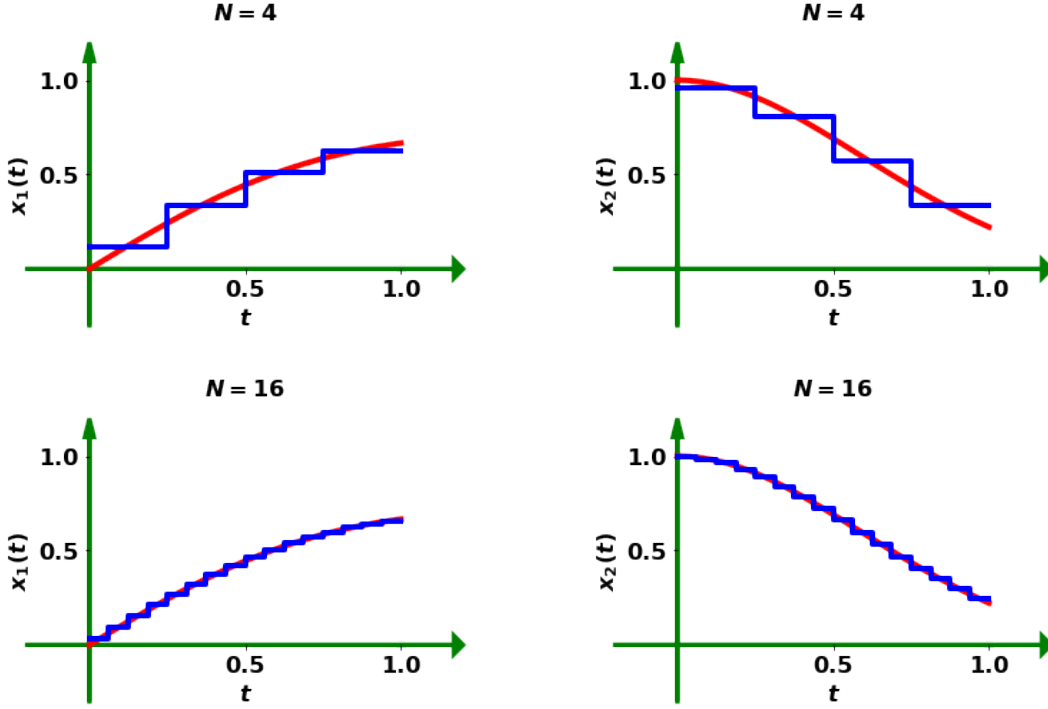
Figure 8: Analytic solution in red for $x_1(t)$ and the solutions obtained by Walsh functions of order $N = 4$ and $N = 16$ (in blue), based on the proposed hybrid classical-quantum approach are shown in the left column. The corresponding results for $x_2(t)$ are shown in the right column.

form for an arbitrary input vector was implemented and successfully tested on the simulation environment on Qiskit (IBM's open source quantum computing platform). The advantage of the proposed hybrid classical-quantum approach for Walsh-Hadamard transform is significantly lower computational complexity $((\mathcal{O}(N))$ operations) in comparison to the classical Fast Walsh-Hadamard transform $(\mathcal{O}(N \log_2(N))$ operations). This speedup could be utilized to achieve a superior hybrid classical-quantum approach for the solution of nonlinear ordinary differential equations. The proposed approach was demonstrated using two computational examples relevant to nonlinear differential equations for cases involving one and two dependent variables. The results based on the proposed hybrid classical-quantum approach for the solution of nonlinear differential equations were found to be encouraging and they matched the results obtained from corresponding classical approaches (and exact solutions).

The relation between Walsh-Hadamard functions and the character theory of finite groups is also explored to provide an alternate and perhaps a more conceptually rigorous approach for constructing Walsh-Hadamard functions. The character theory also provides a natural ordering of Walsh-Hadamard basis functions, compatible with unitary transformations associated with quantum Hadamard gates. In contrast to previous works relevant to classical solutions of ordinary differential equations based on Walsh functions, which mostly focused on sequency ordering, this work presents new formulations and results based on the natural ordering of Walsh-Hadamard basis functions.

Future work involves extensions of the proposed hybrid classical-quantum approach to the solution of partial differential equations and quantum machine learning algorithms. Many classical applications (including signal processing and image compression) where classical Walsh-Hadamard transforms are cur-

rently used can also benefit from the hybrid classical-quantum approach for computing Walsh-Hadamard transform (a key step in the proposed ODE solution approach).

References

[1] Henry F Ahner. Walsh functions and the solution of nonlinear differential equations. American Journal of Physics, 56(7):628–633, 1988.

[2] Kenneth George Beauchamp. Walsh functions and their applications. 1975.

[3] Tom Beer. Walsh transforms. American Journal of Physics, 49(5):466–472, 1981.

[4] Dominic W Berry. High-order quantum algorithm for solving linear differential equations. Journal of Physics A: Mathematical and Theoretical, 47(10):105301, 2014.

[5] Dominic W Berry, Andrew M Childs, Aaron Ostrander, and Guoming Wang. Quantum algorithm for linear differential equations with exponentially improved dependence on precision. Communications in Mathematical Physics, 356(3):1057–1081, 2017.

[6] John P Boyd. Chebyshev and Fourier spectral methods. Courier Corporation, 2001.

[7] John Charles Butcher. Numerical methods for ordinary differential equations. John Wiley & Sons, 2016.

[8] CF Chen and CH Hsiao. A Walsh series direct method for solving variational problems. Journal of the Franklin Institute, 300(4):265–280, 1975.

[9] Andrew M Childs and Jin-Peng Liu. Quantum spectral methods for differential equations. Communications in Mathematical Physics, 375(2):1427–1457, 2020.

[10] William Fulton and Joe Harris. Representation theory: a first course, volume 129. Springer Science & Business Media, 2013.

[11] Youssef A. Geadah and MJG Corinthios. Natural, dyadic, and sequency order algorithms and processors for the Walsh-Hadamard transform. IEEE Transactions on Computers, 26(05):435–442, 1977.

[12] Peter A Gnoffo. Global series solutions of nonlinear differential equations with shocks using Walsh functions. Journal of Computational Physics, 258:650–688, 2014.

[13] Peter A Gnoffo. Unsteady solutions of non-linear differential equations using Walsh function series. In 22nd AIAA Computational Fluid Dynamics Conference, page 2756, 2015.

[14] Peter A Gnoffo. Solutions of nonlinear differential equations with feature detection using fast Walsh transforms. Journal of Computational Physics, 338:620–649, 2017.

[15] David Harvey and Joris Van Der Hoeven. Integer multiplication in time O(n log n). Annals of Mathematics, 193(2):563–617, 2021.

[16] WS Kuklinski. Fast Walsh transform data-compression algorithm: Ecg applications. Medical and Biological Engineering and Computing, 21(4):465–472, 1983.

[17] Henry O. Kunz. On the equivalence between one-dimensional discrete Walsh-Hadamard and multi-dimensional discrete Fourier transforms. IEEE Transactions on Computers, 28(03):267–268, 1979.

[18] Sarah K Leyton and Tobias J Osborne. A quantum algorithm to solve nonlinear differential equations. arXiv preprint arXiv:0812.4423, 2008.

[19] Jin-Peng Liu, Herman Øie Kolden, Hari K Krovi, Nuno F Loureiro, Konstantina Trivisa, and Andrew M Childs. Efficient quantum algorithm for dissipative nonlinear differential equations. Proceedings of the National Academy of Sciences, 118(35), 2021.

[20] Seth Lloyd, Giacomo De Palma, Can Gokler, Bobak Kiani, Zi-Wen Liu, Milad Marvian, Felix Tennie, and Tim Palmer. Quantum algorithm for nonlinear differential equations. arXiv preprint arXiv:2011.06571, 2020.

[21] Yi Lu and Yvo Desmedt. Walsh transforms and cryptographic applications in bias computing. Cryptography and Communications, 8(3):435–453, 2016.

[22] Michael A. Nielsen and Isaac Chuang. Quantum Computation and Quantum Information. Cambridge University Press, 2000.

[23] Jean-Pierre Serre. Linear representations of finite groups, volume 42. Springer, 1977.

[24] Benjamin Steinberg. Representation theory of finite groups: an introductory approach. Springer, 2012.

[25] Endre Süli. Numerical solution of ordinary differential equations. Mathematical Institute, University of Oxford, 2010.

[26] Joseph L Walsh. A closed set of normal orthogonal functions. American Journal of Mathematics, 45(1):5–24, 1923.

[27] C Zarowski and Maurice Yunik. Spectral filtering using the fast Walsh transform. IEEE transactions on acoustics, speech, and signal processing, 33(5):1246–1252, 1985.