Concept Paper

Not peer-reviewed version

Redefining the Mathematical Foundations of Quantum Computing to Significantly Expand the Capability of Quantum Computers

Peter Jean Paul * and shanaz Wahid

Posted Date: 17 September 2025

doi: 10.20944/preprints202509.1320.v1

Keywords: Quantum computing; Semi-structured complex numbers; Pauli matrices; SU(2) matrices

Concept Paper

Redefining the Mathematical Foundations of Quantum Computing to Significantly Expand the Capability of Quantum Computers

Peter Jean-Paul [1,*] and Shanaz Wahid [2]

[1] Ministry of Housing and Local Government, Waterfront, Castries, Saint Lucia.

[2] Mathematics & Statistics, Faculty of Science and Technology, University of the West Indies, St. Augustine, Trinidad and Tobago

* Correspondence: peter.jnpaulnz@gmail.com

Abstract

Quantum computing (defined on the mathematical framework of complex numbers) is limited by its inability to divide by zero. This curtails the ability of quantum computers to handle singularities and infinities in simulations and solve ill-posed problems. This paper aims to define three new but distinct mathematical frameworks for building quantum computing architectures to expand the capabilities of quantum computers by enabling division by zero in quantum calculations. To do this, semi-structured complex numbers were partitioned into 3 distinct subsets of numbers: *i-numbers, p-numbers and k-numbers*. For each subset a novel set of Pauli matrices (representing quantum gates) were derived. The matrices and subsets were used to construct four mathematical Frameworks (111-Framework, 211-Framework, 121-Framework and 112-Framework). To demonstrate the utility of the frameworks the 211-Framework was selected, a universal gate set (from which all other quantum gates and circuits can be built) was defined, a novel Bloch sphere was constructed and the solution to an ill-defined problem was demonstrated. This paper provides a rigorous, and consistent way of incorporating division by zero into quantum computing and extends the computational reach of quantum computers.

Keywords: Quantum computing; Semi-structured complex numbers; Pauli matrices; SU(2) matrices

1. Introduction

Complex numbers, combined with linear algebra, form the core mathematical framework for quantum computing. The complex number set is used to describe qubit states, the unitary transformations (or quantum gates) that manipulate qubit states, and provides quantum computers with a way to describe unique quantum properties (such as phase, interference and unitary evolution) [1,2]. Nevertheless, current quantum computing, based on standard complex numbers, is limited by the inability to divide by zero [3]. Whilst this is a very deep and widely accepted mathematical constraint, this does have direct implications for quantum computing capabilities [4].

For example, computer simulations of physical systems sometimes involve singularities. Particularly in high energy physics, cosmology, or even certain material properties at extreme conditions, can exhibit singularities in mathematical computations used to describe these systems [5,6]. A quantum computer capable of consistently handling division by zero could directly simulate these phenomena [7]. However, this is currently impossible and requires heavy approximation [8].

Additionally, some quantum field theories grapple with infinities [9]. Any number set that can be used to enable division by zero in calculations in a consistent logical way and give meaning to these infinities within a computational context, could unlock simulations of complex quantum systems that are currently beyond reach [10].

Moreover, qubits that represent quantum information do not encode infinities or indeterminate states in a non-contradictory way [11]. A mathematical framework that could be used to create new types of qubits (capable of logically and consistently defining division by zero) could lead to a new way of encoding information beyond the standard basis states [12].

Thirdly, it is often the case that mathematical solutions to complex real-world problems are sometimes not unique or stable because of existing singularities in these solutions [13]. A quantum computer with the capability of handling division by zero could provide a consistent way to find meaningful solutions to such problems [14].

Finally, optimization problems often involve searching for vast landscapes, sometimes with infinite peaks or valleys [15]. If division by zero could navigate these consistently, it might lead to an even more powerful optimization algorithm Particularly in the areas of cryptography, cosmology, logistics, finance, and material design [16].

Division by zero needs to be incorporated into quantum computing to realize the advantages of having this operation on quantum computers. As a starting point, this paper utilizes two tools: (1) semi-structured complex numbers and (2) Pauli matrices.

1.1. Semi-Structured Complex Numbers

Semi-structured complex numbers are numbers that were created specifically to enable division by zero in regular algebraic equations [17]. Semi-structured complex number set can be defined as follows:

A semi-structured complex number is a three-dimensional number of the general form $h = x + yi + zp$; *that is, a linear combination of real* (1), *imaginary* (i) *and unstructured* (p) *units whose coefficients* x, y, z *are real numbers.*

The number h is called semi-structured complex because it contains a structured complex part $(x + yi)$ and an unstructured part (zp). Integer powers of p yield the following cyclic results:

$$p^1 = \frac{1}{0} \quad p^2 = -1 \quad p^3 = -p \quad p^4 = 1 \quad p^5 = \frac{1}{0} \quad p^6 = -1 \quad p^7 = -p \quad \cdots$$

Given the definition of semi-structured complex numbers, it can clearly be seen that infinity (represented by division by zero) is encoded within the number and can be dealt with algebraically. Other important characteristics of semi-structured complex numbers is given in Table A1 in Appendix A1.

Semi-structured complex numbers can be seen as vectors in a 3-dimensional Euclidean semi-structured complex space [18]. This representation enables vector operations (such as addition, dot product, inner product) to be performed on semi-structured complex numbers [19]. This implies that semi structured complex numbers can be used within linear algebra, the algebra of quantum computing [20].

The geometric properties of semi structured complex numbers as an extension of complex numbers make them ideal for being used within quantum computing. Since "p" behaves like "i" in that it has higher order cyclic behavior, it is possible to build quantum gates, quantum states, and protocols using the semi-structured complex number set.

1.2. Pauli Matrices

1.2.1. Origin and Meaning of the Pauli Matrices

Another tool that is necessary for the introduction of division by zero into quantum computing is the Pauli matrices [21]. The Pauli matrices is a set of three 2×2 complex matrices that were created by physicist Wolfgang Pauli who introduced them as part of his work on the theory of spin in quantum mechanics [22].

As a bit of background, in the early 20th century, the Stern–Gerlach experiment showed that electrons possess intrinsic angular momentum (spin) that can only take certain discrete values [23].

However, traditional quantum mechanics (based on Schrödinger's equation) couldn't fully explain this property. Traditional quantum mechanics was formulated using wave mechanics. In wave mechanics, the basis vectors for angular momentum are $S_x = (1,0,0)$, $S_y = (0,1,0)$ and $S_z = (0,0,1)$. Since traditional quantum mechanics could not fully explain the property of spin, a mathematically equivalent form of quantum mechanics had to be used. This form of quantum mechanics was called matrix mechanics.

Matrix mechanics proved very efficient in describing spin [24]. With matrix mechanics the basis vectors for angular momentum were converted into an equivalent basis matrix form [25]. These equivalent matrices were called Pauli matrices. The Pauli matrices have property of being Hermitian (meaning that the matrix is equal to its conjugate transpose) and unitary (the inverse of the matrix is equal to its conjugate transpose) [26]. To convert a 3-dimensional vector $v = (v_1, v_2, v_3)$ into a 2×2 Hermitian matrix, the following rule is applied:

$$\boldsymbol{V} = \begin{bmatrix} v_3 & v_1 + iv_2 \\ v_1 - iv_2 & -v_3 \end{bmatrix} \tag{1}$$

Hence, the basis vector along the x-direction $S_x = (1 \quad 0 \quad 0)$ can be written in matrix form as

$$S_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{2}$$

Similarly, the basis vector along the y-direction $S_y = (0 \quad 1 \quad 0)$ can be written in matrix form as:

$$S_y = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix} \tag{3}$$

And finally, the basis vector along the z-direction $S_z = (0 \quad 0 \quad 1)$ can be written in matrix form as:

$$S_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{4}$$

In quantum computing the Pauli matrices are used as fundamental quantum logic gates acting on qubits. In that form they are given new symbols and new names. The Pauli matrices used in quantum computing are given in Table 1.

Table 1. Pauli matrices as fundamental quantum logic gates.

Gate	Name	Description
$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	Pauli-X matrix or X-gate	It is called the bit-flip operator. It essential flips a qubit to its opposite. That is, changes $\|0\rangle$ to $\|1\rangle$ and $\|1\rangle$ to $\|0\rangle$. On the Bloch sphere, the X gate corresponds to a 180° rotation around the x-axis.
$\sigma_y = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}$	Pauli-Y matrix or Y-gate	Pauli-Y flips the qubit and multiplies by $\pm i$. That is, changes $\|0\rangle$ to $i\|1\rangle$ and $\|1\rangle$ to $-i\|0\rangle$. On the Bloch sphere, the Y gate corresponds to a 180° rotation around the y-axis.
$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	Pauli-Z matrix or Z-gate	Pauli-Z Performs a phase flip on a qubit. That is, it leaves $\|0\rangle$ unchanged and changes $\|1\rangle$ and $-\|1\rangle$. On the Bloch sphere, the Z gate performs a 180° rotation around the z-axis.

Any single qubit state can be represented as a linear combination of the Pauli matrices and the identity matrix [27]. This is part of the formalism used to describe qubits in quantum computing. Pauli matrices are also considered SU(2) matrices. SU(2) in this case means "special unitary 2×2 matrices"; special means that the determinant of these matrices is 1 and unitary means that the conjugate transpose of the matrix is the same as its inverse [28].

1.2.2. Using the Pauli Matrices to Construct a Bloch Sphere

Table 1 makes mention of Bloch sphere. A Bloch sphere is simply a visual representation of the state of a qubit as shown in Figure 1.

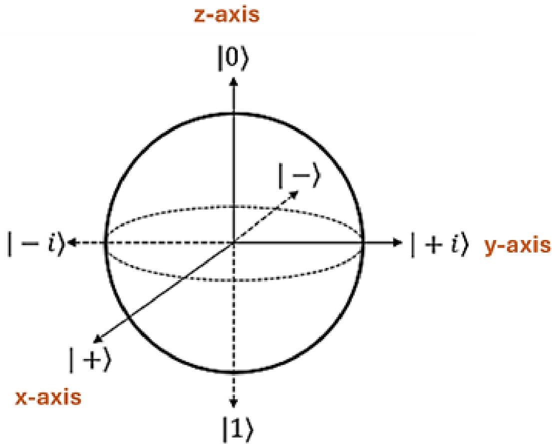

Figure 1. Bloch sphere describing qubits |0⟩ and |1⟩.

What is critical to know is how the Bloch sphere is derived from the Pauli matrices [29]. The Bloch Sphere consists of three axes: x-axis, y-axis, and z-axis. The values and directions of each axis is constructed from the eigenstates and eigenvalues of the corresponding Pauli matrix [30]. The relationship between the Pauli matrices and the construction of the Bloch sphere is given in Table 2.

Table 2. Relationship between Pauli Matrices and Bloch Sphere.

Eigenstates and Eigenvalues for Pauli Matrices	Use in constructing the Bloch Sphere
Z-axis	
$\begin{pmatrix}1\\0\end{pmatrix}$ is the first eigenstate of σ_z with eigenvalue +1.	On the z-axis $\begin{pmatrix}1\\0\end{pmatrix}$ is represented as |0> and the eigen value +1 indicates that |0> is on the positive z-axis.
$\begin{pmatrix}0\\1\end{pmatrix}$ is the second eigenstate of σ_z with eigenvalue −1.	On the z-axis $\begin{pmatrix}0\\1\end{pmatrix}$ is represented as |1> and the eigenvalue −1 indicates that |1> is on the negative z-axis.
X-axis	
$\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$ is the first eigenstate of σ_x with eigenvalue +1.	On the x-axis $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} = \frac{1}{\sqrt{2}}(\|0> + \|1>)$ is represented as |+> and the eigenvalue +1 indicates that |+> is on the positive x-axis.
$\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$ is the second eigenstate of σ_x with eigenvalue −1.	On the x-axis $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix} = \frac{1}{\sqrt{2}}(\|0> - \|1>)$ is represented as |−> and the eigenvalue −1 indicates that |−> is on the negative x-axis.
Y-axis	
$\frac{1}{\sqrt{2}}\begin{pmatrix}1\\i\end{pmatrix}$ is the first eigenstate of σ_y with eigenvalue +1.	On the x-axis $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\i\end{pmatrix} = \frac{1}{\sqrt{2}}(\|0> + i\|1>)$ is represented as |+i> and the eigenvalue +1 indicates that |+i> is on the positive y-axis.
$\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-i\end{pmatrix}$ is the second eigenstate of σ_y with eigenvalue −1.	On the x-axis $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-i\end{pmatrix} = \frac{1}{\sqrt{2}}(\|0> - i\|1>)$ is represented as |−i> and the eigenvalue −1 indicates that |−i> is on the negative y-axis.

Table 2 highlights the fact that there's a very strong relationship between the eigenstates and eigenvalues of the Pauli matrices and the construction of the Bloch Sphere used to describe qubits.

1.3. Major Contributions

Given a basic understanding of quantum computing, semi-structured complex numbers, and Pauli matrices, the aim of this paper is to:

Use semi structured complex numbers to define three distinct number sets that can enable division by zero and simultaneously form four core foundational yet distinct mathematical frameworks for building quantum computers.

In the process of achieving this aim the following major contributions are made:

1. Using the 4D form of semi structured complex numbers (that is: $h = a + bi + cp + dk$) to create three new number sets that are subsets of semi-structured complex numbers. These number sets are given in Table 3.

Table 3. New number sets.

Number set	Name
$a + bi$	i–*numbers* (Complex numbers)
$a + cp$	p–*numbers*
$a + dk$	k–*numbers*

Note: where a, b, c, d are real constants, i and p are the imaginary and unstructured units respectively and $k = ip$

2. Use the new number subsets to define the Generalized Pauli Matrix Set consisting of six distinct special unitary matrices. The Generalized Pauli Matrix Set is given in Table 4.

Table 4. Generalized Pauli Matrix Set.

	X	Y	Z
1	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
2	$\begin{pmatrix} 0 & k \\ k & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & p \\ -p & 0 \end{pmatrix}$	$\begin{pmatrix} k & 0 \\ 0 & -k \end{pmatrix}$

Each matrix in the table is named according to the column and row that it belongs to. For example, $X_2 = \begin{pmatrix} 0 & k \\ k & 0 \end{pmatrix}$ (pronounced "Pauli X two"). Addtionally, $Z_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ and so on. The positioning of these matrices in table is not arbitrary but is based on the method used to arrive at them; that is, the position of the matrices cannot change.

Using the number sets from Contribution 1 and the Generalized Pauli Matrix Set from Contribution 2 the following four mathematical frameworks were created. The frameworks are given in Table 5. What differentiates each framework is the matrices used to create the logic gates for quantum computing and mathematical number sets that underline each framework.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 September 2025 doi:10.20944/preprints202509.1320.v1

Table 5. Foundational frameworks for quantum computing.

Name of Framework	Number set used within Framework	Pauli matrices used to build quantum gates within Framework
111–Framework	*i–numbers*	$X_1 Y_1 Z_1$
211–Framework	*k–numbers*	$X_2 Y_1 Z_1$
121–Framework	*p–numbers*	$X_1 Y_2 Z_1$
112–Framework	*k–numbers*	$X_1 Y_1 Z_2$

Note:
1. *The numbers appearing in the names of the frameworks are based on and match the subscripts of the matrices used to form the logic gates for these frameworks*
2. *The* 2×2 *identity matrix* ***I*** *forms the 4th matrix used in every framework.*

Each of these frameworks offer distinct computational advantages. The last three frameworks enable division by zero within computation as well as within quantum logic circuits.

3. For each framework outlined in Table 5, a universal gate set (from which all other quantum gates and circuits can be built) was developed and proved to adhere to the Solovay Kitaev theorem. The theorem states that: "*if a set of single-qubit gates can generate a dense subgroup of SU(2), then any desired single-qubit gate can be approximated to an arbitrary precision* ϵ *using a sequence of gates from this finite set whose length scales* poly-nominally in $log\left(\frac{1}{\epsilon}\right)$". The universal gift set under each mathematical framework is given in Table 6.

Table 6. Universal Gate Sets for each quantum computing mathematical framework.

Name of Framework	Universal Gate Set	Matrix representation of gates within Universal Gate Set
111–Framework	$\{H, T\}$	$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$
211–Framework	$\{H, T_k\}$	$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ $T_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{k\frac{\pi}{4}} \end{pmatrix}$
121–Framework	$\{H, T_p\}$	$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ $T_p = \begin{pmatrix} 1 & 0 \\ 0 & e^{p\frac{\pi}{4}} \end{pmatrix}$
112–Framework	$\{H, T_{-k}\}$	$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ $T_{-k} = \begin{pmatrix} 1 & 0 \\ 0 & e^{-k\frac{\pi}{4}} \end{pmatrix}$

Note:
1. H is called the Hadamar gate, T is called the T-gate.
2. For each gate set the inverses of the gates in the set are implicitly included as a sequence of gates can always be inverted

4. As an example, the utility of one of the frameworks 211-Framework was demonstrated.

The rest of the paper is devoted to showing how in the process of achieving the aim the four major contributions were derived.

2. Defining Four Distinct Mathematical Frameworks of Quantum Computing

2.1. Characterizing the Three Distinct Subsets of Semi-Structured Complex Numbers

The fundamental representation of semi structured complex numbers is given in Equation (5).

$$h = a + bi + cp + dip \quad (5)$$

where	
a, b, c, d	real constants
i	imaginary unit
p	Unstructured unit

Equation (5) can be rewritten as

$$h = a + bi + cp + dk \quad (5)$$

where $k = ip$.

Semi structured complex numbers can be divided into 3 distinct subsets of numbers. These subsets are given in Table 7.

Table 7. New number sets.

Number set	Name
$a + bi$	*i–numbers* (Complex numbers)
$a + cp$	*p–numbers*
$a + dk$	*k–numbers*

Note: where a, b, c, d are real constants, i and p are the imaginary and unstructured units respectively and $k = ip$

Quantum computing is based on linear algebra. For a number set to be used within linear algebra it must satisfy 11 field axioms [31]. Field axioms are a set of fundamental rules that define a mathematical structure called a field. In abstract algebra, a field is a set on which addition, subtraction, multiplication, and division (except by zero) are defined and behave in a way that allows us to do arithmetic in a consistent and predictable way. If a set with two operations (usually called "addition" and "multiplication") satisfies all these axioms, then it's a field. These 11 field axioms and proof that each number set obeys these axioms are given in Tables A2 and A3 in Appendix A2. Additionally, whilst these number sets share real numbers as common elements, what makes the number sets distinct is that no one number set can be used on its own to fully derive another number set [32].

2.2. Characterizing the "Generalized Pauli Matrix Set"

To define the different mathematical frameworks for quantum computing the next step was to consider the possible Pauli type matrices that can be formed from the number sets defined in given in Table 7.

Table 7. New number sets.

Number set	Name
$a + bi$	*i–numbers* (Complex numbers)
$a + cp$	*p–numbers*
$a + dk$	*k–numbers*

Note: where a, b, c, d are real constants, i and p are the imaginary and unstructured units respectively and $k = ip$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 September 2025 doi:10.20944/preprints202509.1320.v1

It is necessary to demonstrate that each of these number sets can be used to create Pauli type matrices because it is those matrices that are necessary to represent physical quantities in quantum computing and also to create quantum logic gates and consequently quantum logic circuits for quantum computers.

Therefore, these Pauli type matrices for each number set must be both Hermitian and Unitary. Hermitian matrices are essential for representing observables (physical quantities that can be measured) and unitary matrices are essential for representing quantum gates (operations on qubits). Matrices that carry both of these properties can be used within quantum computing for representing superposition, interference, and error correction and stabilization codes for single-qubit and multi-qubit gate operations and measurements.

When handling the complex number set (the i-numbers) it was found that to convert 3D-vectors (v_1, v_2, v_3) into 2×2 Pauli matrices the following formulation is usually used:

$$\boldsymbol{T_1} = \begin{bmatrix} v_3 & v_1 + iv_2 \\ v_1 - iv_2 & -v_3 \end{bmatrix} \tag{6}$$

The transformation $\boldsymbol{T_1}$ is a complex number (or i–numbers) based matrix; this can easily be seen from the top right hand corner element in the matrix which has the complex number form $v_1 + iv_2$. Therefore, the transformation for the basis vectors $(1 \quad 0 \quad 0)$, $(0 \quad 1 \quad 0)$, $(0 \quad 0 \quad 1)$ into 2×2 matrices using transformation $\boldsymbol{T_1}$ gives:

$$X_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad Y_1 = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix} \qquad Z_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{7}$$

$X_1 Y_1 Z_1$ are the Pauli matrices. As stated earlier these matrices are both Hermitian and Unitary. These matrices along with the identity matrix are some of the matrices used to form quantum logic gates for the current quantum computing paradigm.

However, the transformation $\boldsymbol{T_1}$ is not the only matrix transformation that can be used to create 2×2 Hermitian unitary matrices. Using the p-numbers and k-numbers, seven other transformation matrices can be created. The complete set of transformation matrices that can be created from the three number sets i-numbers, p-numbers and k- numbers is given in Table 8.

Table 8. Transformations (using new number sets) used to create 2 × 2 Hermitian unitary matrices.

$\boldsymbol{T_1} = \begin{bmatrix} v_3 & v_1 + v_2 i \\ v_1 - v_2 i & -v_3 \end{bmatrix}$	$\boldsymbol{T_2} = \begin{bmatrix} v_3 & v_1 k + v_2 i \\ v_1 k - v_2 i & -v_3 \end{bmatrix}$	$\boldsymbol{T_3} = \begin{bmatrix} v_3 k & v_1 + v_2 i \\ v_1 - v_2 i & -v_3 k \end{bmatrix}$	$\boldsymbol{T_4} = \begin{bmatrix} v_3 k & v_1 k + v_2 i \\ v_1 k - v_2 i & -v_3 k \end{bmatrix}$
$\boldsymbol{T_5} = \begin{bmatrix} v_3 & v_1 + v_2 p \\ v_1 - v_2 p & -v_3 \end{bmatrix}$	$\boldsymbol{T_6} = \begin{bmatrix} v_3 & v_1 k + v_2 p \\ v_1 k - v_2 p & -v_3 \end{bmatrix}$	$\boldsymbol{T_7} = \begin{bmatrix} v_3 k & v_1 + v_2 p \\ v_1 - v_2 p & -v_3 k \end{bmatrix}$	$\boldsymbol{T_8} = \begin{bmatrix} v_3 k & v_1 k + v_2 p \\ v_1 k - v_2 p & -v_3 k \end{bmatrix}$

Proof that the transformations in Table 8 form 2×2 Hermitian unitary matrices is given in Appendix A3. Each transformation matrix in Table 8 was used to convert each basis vector $(1 \quad 0 \quad 0)$, $(0 \quad 1 \quad 0)$, $(0 \quad 0 \quad 1)$ into Hermitian unitary matrices. The result is given in Table 9. Table 9 has 24 entries. The entries highlighted in grey are the unique entries of the table that span the entire table. Every other entry is a copy of one of the entries highlighted in grey. This observation means that Table 9 can be greatly simplified.

Table 9. Resulting 2 × 2 Hermitian unitary matrices from applying transformations T_1 to T_8 on 3D basis vectors.

Transformations	Basis Vectors X $(1\ 0\ 0)$	Basis Vectors Y $(0\ 1\ 0)$	Basis Vectors Z $(0\ 0\ 1)$
T_1	$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$	$\begin{bmatrix}0 & i\\-i & 0\end{bmatrix}$	$\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}$
T_2	$\begin{bmatrix}0 & k\\k & 0\end{bmatrix}$	$\begin{bmatrix}0 & i\\-i & 0\end{bmatrix}$	$\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}$
T_3	$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$	$\begin{bmatrix}0 & i\\-i & 0\end{bmatrix}$	$\begin{bmatrix}k & 0\\0 & -k\end{bmatrix}$
T_4	$\begin{bmatrix}0 & k\\k & 0\end{bmatrix}$	$\begin{bmatrix}0 & i\\-i & 0\end{bmatrix}$	$\begin{bmatrix}k & 0\\0 & -k\end{bmatrix}$
T_5	$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$	$\begin{bmatrix}0 & p\\-p & 0\end{bmatrix}$	$\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}$
T_6	$\begin{bmatrix}0 & k\\k & 0\end{bmatrix}$	$\begin{bmatrix}0 & p\\-p & 0\end{bmatrix}$	$\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}$
T_7	$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$	$\begin{bmatrix}0 & p\\-p & 0\end{bmatrix}$	$\begin{bmatrix}k & 0\\0 & -k\end{bmatrix}$
T_8	$\begin{bmatrix}0 & k\\k & 0\end{bmatrix}$	$\begin{bmatrix}0 & p\\-p & 0\end{bmatrix}$	$\begin{bmatrix}k & 0\\0 & -k\end{bmatrix}$

To simplify Table 9, columns X, Y, Z are kept. Each column has two highlighted unique matrices; two unique X matrices, two unique Y matrices and two unique Z matrices. Using this observation Table 9 can be condensed to give Table 10.

Table 10. The Generalized Pauli Matrix Set.

	Ⓧ	Ⓨ	Ⓩ
①	$\begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}$	$\begin{pmatrix}0 & i\\-i & 0\end{pmatrix}$	$\begin{pmatrix}1 & 0\\0 & -1\end{pmatrix}$
②	$\begin{pmatrix}0 & k\\k & 0\end{pmatrix}$	$\begin{pmatrix}0 & p\\-p & 0\end{pmatrix}$	$\begin{pmatrix}k & 0\\0 & -k\end{pmatrix}$

Table 10 is called the Generalized Pauli Matrix Set. Each matrix in the table is named according to the column and row that it belongs to. For example, $X_2 = \begin{pmatrix}0 & k\\k & 0\end{pmatrix}$ (pronounced "Pauli X two"), $Z_1 = \begin{pmatrix}1 & 0\\0 & -1\end{pmatrix}$ and so on. The positioning of the matrices in table is not arbitrary but is based on the method used to arrive at them. Therefore, the position of the matrices does not change.

2.3. Characterizing the Foundational Mathematical Frameworks

Using the number sets in Table 7 and the Generalized Pauli Matrix Sets from Table 10, four mathematical frameworks were created. These frameworks are given in Table 11.

Table 11. Foundational Mathematical frameworks for quantum computing.

Name of Framework	Number set used within Framework	Pauli matrices used to build quantum gates within Framework
111–Framework	*i–numbers*	$X_1Y_1Z_1$
211–Framework	*k–numbers*	$X_2Y_1Z_1$
121–Framework	*p–numbers*	$X_1Y_2Z_1$
112–Framework	*k–numbers*	$X_1Y_1Z_2$

Note:
1. *The numbers appearing in the names of the frameworks are based on and match the subscripts of the matrices used to form the logic gates for these frameworks*
2. *The 2×2 identity matrix **I** forms the 4th matrix used in every framework.*

What differentiates each framework is the matrices used to create the logic gates for quantum computing and mathematical number sets that underline each framework. Each of these frameworks offer distinct computational advantages. For example, the last three frameworks enable division by zero within computation as well as within quantum logic circuits.

Each mathematical framework has a unique set of logic gates that can be to create quantum logic circuits and quantum algorithms. Because each mathematical framework is distinct, quantum computers based on one framework are fundamentally distinct from quantum computers based on another framework. Nevertheless, all quantum computing (irrespective of the mathematical framework used to underline its operation) will still operate on the basic principles of linear algebra.

Since the foundations of quantum computing is well established for the 111–Framework (that is for complex numbers), the basic foundations of the other frameworks, that is, the 211–Framework, 121–Framework, and 112–Framework needs to be examined. Since it is a bit cumbersome to examine all three frameworks within one paper, it suffices to examine one of these frameworks as an example of how the other two frameworks can be developed and integrated within quantum computing. The 211-Framework was chosen to be examined.

3. Foundations of Quantum Computing with 211–Framework

3.1. Core Axioms and Structural Properties

The basic characteristic of the 211–Framework is given in Table 12.

Table 12. 211-Framework for quantum computing.

Name of Framework	Number set used	Quantum gates within Framework
211–Framework	*k–numbers*	$X_2Y_1Z_1$

*Note: The 2×2 identity matrix **I** forms the 4th matrix used in every framework.*

The elements in the k-number set can be defined as

$$k\text{–}numbers = \{a + dk | a, d \in R\} \tag{8}$$

According to the field axioms proofs given in Appendix A1, k-number set is closed under multiplication and addition and contains the identity element 1 and a nontrivial element $k = ip = \frac{\sqrt{-1}}{0}$. The multiplication rules for the k-number set are commutative and 2-cyclic as shown in Table 13.

Table 13. Multiplication Table for *k-numbers*.

$\times$	1	k	k^2	k^3
1	1	k	1	k
k	k	1	k	1
k^2	1	k	1	k
k^3	k	1	k	1

In addition to a well-defined multiplication table, the k–number has a well-defined conjugate; that is, $\overline{a+dk} = a - dk$. Additionally, if x is a k–number then the norm of x is:

$$\text{Norm: } \|x\|^2 = x\bar{x} \in \mathbb{R} \tag{9}$$

3.2. k-Qubits and Vector Spaces

As a basic element the k–qubit is defined over *k–numbers* as Equation (11).

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad \text{where} \qquad \alpha, \beta \in k\text{–}numbers \tag{10}$$

k-qubit states are written using Dirac notation where $|0\rangle$ and $|1\rangle$ represent the "basis states" (analogous to 0 and 1). The equation above represents the superposition between the basis states. Normalization (that is, the magnitude (or length) of the state vector) must be equal to 1. In terms of the inner product this given as:

$$\langle\psi|\psi\rangle = \bar{\alpha}\alpha + \bar{\beta}\beta = 1 \qquad \text{where} \qquad \alpha, \beta \in k\text{–}numbers \tag{11}$$

Superpositions as shown in Equation (11) cannot be expressed in terms of complex numbers (that is, *i–numbers*), allowing a richer spectrum of interference. These k-qubits suggest a "supra-quantum" theory; that is, physical theory that is "stronger" or "more general" than standard quantum mechanics yet still respects some fundamental physical principles that quantum mechanics itself adheres to. This expands the structure of superposition and enabling new classes of computational or physical states.

3.3. k-Pauli Operators as Building Blocks for 211–Framework of Quantum Computing

Since the 211–Framework of quantum computing is considered, based on Table 11, the gates that would be used in this framework is $X_2Y_1Z_1$. This is given in Equation (13).

$$X_2 = \begin{pmatrix} 0 & k \\ k & 0 \end{pmatrix} \qquad Y_1 = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix} \qquad Z_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{12}$$

These gates are considered Pauli matrices within the 211–Framework of quantum computing; that is Pauli-X_2, Pauli-Y_1, Pauli-Z_1. These quantum gates are the operations that manipulate k-qubit states. They are the quantum equivalent of logic gates in classical computing.

In quantum mechanics there is an operation called the commutator operator. This operation gives the extent to which two operators (or matrices) can be measured simultaneously with arbitrary precision. This is given by the equation $[A, B] = AB - BA$ where A and B are operators or matrices. If the final value of the commutator is zero, then this means operators A and B can be measured simultaneously with arbitrary precision. The $X_2Y_1Z_1$ gates satisfy the following commutator properties shown in Result (14).

$$[X_2, Y_1] = 2pZ_1 \qquad [Y_1, Z_1] = 2pX_2 \qquad [Z_1, X_2] = 2pY_1 \tag{13}$$

Where p is the unstructured unit p. Proof of these properties is given in Appendix A3.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 September 2025 doi:10.20944/preprints202509.1320.v1

The Pauli matrices outlined here are "unitary rotations"; that is, transformations that rotate the quantum state vector around in its Hilbert space, without changing its length. This "rotation" changes the probability amplitudes and phases of the basis states. The amount of rotation produced by each gate is given in Table 14.

Table 14. Fundamental quantum logic gates in 211–Framework.

Gate	Name	Description
$X_2 = \begin{pmatrix} 0 & k \\ k & 0 \end{pmatrix}$	Pauli-X_2 matrix or X_2-gate	It is called the bit-flip operator. It essential flips a qubit to its opposite. That is, changes $\|0\rangle$ to $\|1\rangle$ and $\|1\rangle$ to $\|0\rangle$. On the Bloch sphere, the X_2 gate corresponds to a 180° rotation around the x-axis.
$Y_1 = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}$	Pauli-Y_1 matrix or Y_1-gate	Pauli-Y_1 flips the qubit and multiplies by $\pm i$. That is, changes $\|0\rangle$ to $i\|1\rangle$ and $\|1\rangle$ to $-i\|0\rangle$. On the Bloch sphere, the Y_1 gate corresponds to a 180° rotation around the y-axis.
$Z_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	Pauli-Z_1 matrix or Z_1-gate	Pauli-Z_1 Performs a phase flip on a qubit. That is, it leaves $\|0\rangle$ unchanged and changes $\|1\rangle$ and $-\|1\rangle$. On the Bloch sphere, the Z_1 gate performs a 180° rotation around the z-axis.

The gates listed in Table 14 are called Standard single qubit gates. These single qubit gates are fundamental as they are used to build and or manipulate other gates such as Rotation gates, Pauli-power gates, Quarter turns and Hadamard gates. For example, the three Pauli-matrices can be used to create unitary Pauli-rotation matrices R_{X_2}, R_{Y_1} and R_{Z_1} that rotate a state vector by an arbitrary angle about the corresponding axis in the Bloch sphere. The three unitary rotations are shown below in Equation (15) to Equation (17).

$$R_{X_2}(\theta) = e^{-k\theta X_2} = I\cos(\theta) - k\sin(\theta)\, X_2 \tag{15}$$

$$R_{Y_1}(\theta) = e^{-k\theta Y_1} = I\cos(\theta) - k\sin(\theta)\, Y_1 \tag{16}$$

$$R_{Z_1}(\theta) = e^{-k\theta Z_1} = I\cos(\theta) - k\sin(\theta)\, Z_1 \tag{17}$$

Here I is the identity matrix.

3.4. The k-Bloch Sphere

Similar to Table 2, the eigenstates and eigenvalues of the Pauli matrices of the 211–Framework can be used to construct a unique Bloch sphere for the mathematical framework from which qubits can be adequately described. The eigenstates and eigenvalues for the Pauli matrices of the 211-Framework and how they used to construct a novel k-Bloch Sphere for the framework is given in Table 15.

 doi:10.20944/preprints202509.1320.v1

Table 15. Relationship between Pauli Matrices in the 211–Framework and Bloch Sphere.

Eigenstates and Eigenvalues for Pauli Matrices	Use in constructing the Bloch Sphere
Z-axis	
$\begin{pmatrix}1\\0\end{pmatrix}$ is the first eigenstate of Z_1 with eigenvalue +1.	On the z-axis $\begin{pmatrix}1\\0\end{pmatrix}$ is represented as |0> and the eigen value +1 indicates that |0> is on the positive z-axis.
$\begin{pmatrix}0\\1\end{pmatrix}$ is the second eigenstate of Z_1 with eigenvalue −1.	On the z-axis $\begin{pmatrix}0\\1\end{pmatrix}$ is represented as |1> and the eigenvalue −1 indicates that |1> is on the negative z-axis.
X-axis	
$\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$ is the first eigenstate of X_2 with eigenvalue $+k$.	On the x-axis $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} = \frac{1}{\sqrt{2}}(\|0> + \|1>)$ is represented as $k\|+>$ and the eigenvalue +1 indicates that $k\|+>$ is on the positive x-axis.
$\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$ is the second eigenstate of X_2 with eigenvalue $-k$.	On the x-axis $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix} = \frac{1}{\sqrt{2}}(\|0> - \|1>)$ is represented as $k\|->$ and the eigenvalue $-k$ indicates that $k\|->$ is on the negative x-axis.
Y-axis	
$\frac{1}{\sqrt{2}}\begin{pmatrix}1\\i\end{pmatrix}$ is the first eigenstate of Y_1 with eigenvalue +1.	On the x-axis $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\i\end{pmatrix} = \frac{1}{\sqrt{2}}(\|0> + i\|1>)$ is represented as $\|+i>$ and the eigenvalue +1 indicates that $\|+i>$ is on the positive y-axis.
$\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-i\end{pmatrix}$ is the second eigenstate of Y_1 with eigenvalue −1.	On the x-axis $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-i\end{pmatrix} = \frac{1}{\sqrt{2}}(\|0> - i\|1>)$ is represented as $\|-i>$ and the eigenvalue −1 indicates that $\|-i>$ is on the negative y-axis.

The visual representation of this Bloch sphere is shown in Figure 2.

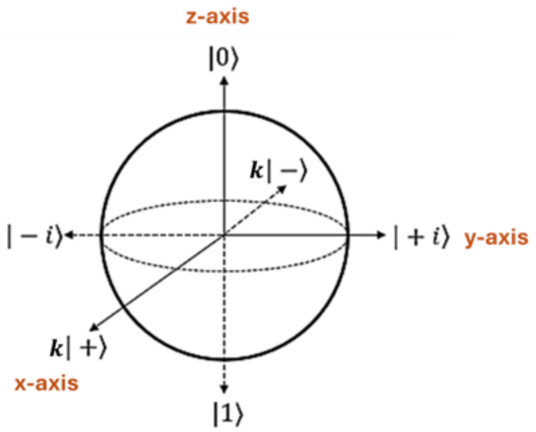

Figure 2. Bloch sphere describing qubits |0⟩ and |1⟩ in the 211-Framework.

The only major change to the original block sphere is the factor of k that appears in the x-axis. This is a direct consequence of the fact that the eigenvalues of the X_2 matrix are multiplied by a factor of k. This factor of k means that the k-Bloch Sphere has a new dimension that does not exist with the original Bloch sphere used in current quantum computing.

An example of where such a difference is important is in the use of density matrices. In quantum computing and quantum mechanics, a density matrix (or density operator) is a matrix used in calculating the probabilities of the outcomes of measurements performed on physical systems. Density matrices are crucial tools in quantum computing that deal with mixed states used to derive quantum information.

A qubit's density matrix ρ_k can be represented on the Bloch sphere using a formula that connects it to a real-valued vector (a, b, c), called the Bloch vector. The density matrix is expressed

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 September 2025 doi:10.20944/preprints202509.1320.v1

as a linear combination of the identity matrix and the k-Pauli matrices, with coefficients from the Bloch vector. This is shown in Equation (18).

$$\rho_k = \frac{1}{2}(I + aX_2 + bY_1 + cZ_1) \qquad \text{where} \qquad a, b, c \in R \qquad (14)$$

Equation (18) generates a k-Bloch sphere, where qubits can be used to explore dimensions that are not used in current quantum computing. This gives the potential to model exotic quantum systems such as extended quantum gravity structures.

3.5. Universal Gate Set and the Solovay-Kitaev Theorem

Sections 3.1 to Section 3.3 shows that the k-numbers behave in a consistent way; that is, the rules and operations defined for them behave in a predictable and logical manner, adhering to the established properties of arithmetic and quantum computing.

It is now necessary to show that a "universal" gate sets exist for the 211-Framework. In classical computing, any logic circuit can be built using a small set of basic logic gates (combination of AND, OR, and NOT gates). This set of logic gates is considered "universal" because any known classical logic circuit can be constructed from these gates. In a similar manner, in quantum computing, for a mathematical framework to be considered fully capable of providing the necessary backbone to build a quantum computing framework it is necessary to show that a set of universal quantum gates exist from which a quantum logical circuits can be built. Since quantum operations are represented by unitary matrices (which in turn represent quantum gates), a universal gate set means that *any unitary matrix* can approximated by combining gates from within that universal gate set. Therefore, we formally define the universal gate set as:

A collection of quantum gates that, when combined in sequences, can approximate any arbitrary unitary transformation on any number of qubits to an arbitrary degree of accuracy.

To show that a set of gates form a universal gate set, it is necessary to show that this set of gates can be used to generate a dense subgroup of SU(2). Here "dense subgroup" means that by combining gates from the finite universal gate set it is possible to get arbitrarily close to any possible single-qubit unitary operation. Note that a "single-qubit unitary operation" is another way of saying a quantum gate.

As a reminder, SU(2) means "special unitary 2×2 matrices" that represent single-qubit unitary operations. Special means that the determinant of these matrices is 1. And unitary means that the conjugate transpose of the matrix is the same as its inverse.

In the 111–Framework (regular quantum computing that uses complex numbers) there are several types of universal gate sets, the simplest of which is the Hadamard gate (H) and the T-gate (T) (and its inverses, which are implicitly included as a sequence of gates can always be inverted). These gates are defined as:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix} \qquad (15)$$

The Hadamard gate creates superpositions and rotates qubits in the Bloch sphere. For example, it maps $|0\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|1\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. The T-gate is a phase shift gate.

The analogues of these gates were used to represent the universal gate set for in the 211–Framework. For the mathematical 211–Framework the following universal gate set is defined: $\{H, T_k\}$ where:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad T_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{k\frac{\pi}{4}} \end{pmatrix} \qquad (16)$$

The universal gate set consist of the Hadamard gate, the T_k–gate. It is of course necessary to prove that $\{H, T_k\}$ forms a universal gate set. The proof is provided in Appendix A5. Another important concept (mentioned in Appendix A5) in defining the universal gate set is the Solovay-Kitaev theorem which is a fundamental result in quantum computation that addresses the problem of approximating arbitrary unitary operations (quantum gates) using a finite set of universal gates [33]. It states that: "*if a set of single-qubit gates can generate a dense subgroup of SU(2), then any desired single-qubit gate can be approximated to an arbitrary precision* ϵ *using a sequence of gates from this finite set whose length scales poly-nominally in* $log\left(\frac{1}{\epsilon}\right)$" [34]. More precisely, the length of the sequence is $O\left(\mathrm{c} \times \log\left(\frac{1}{\epsilon}\right)\right)$ for some constant c. The proof also provides an efficient classical algorithm to find such a sequence.

The theorem is one of the cornerstones of quantum computing demonstration that while the space of quantum operations is continuous, we can efficiently approximate any operation using discrete finite set of basic building blocks, which is essential for building scalable and practical quantum computers [35]. Demonstrating that a universal gate set forms a "dense subset" (as was done in Appendix A5) is sufficient to show that the gate set adheres to the Solovay-Kitaev theorem [36]. This in turn is enough to demonstrate that the mathematical 211-Framework is enough to provide a fully consistent and capable backbone upon which to build quantum computing [37].

3.6. Demonstrating the Utility of the 211-Framework

To demonstrate the utility of the 211-Framework imagine there a quantum algorithm that, at a certain step, needs to apply a Z-rotation (phase shift) to a qubit. The *angle* of this Z-rotation (ϕ) is determined by the "distance" or "overlap" between two previously measured classical values, A and B.

Suppose the angle is defined as: $\phi = \frac{C}{A-B}$, where C is some non-zero constant (e.g., $C = \frac{\pi}{2}$).

Under the 211-Framework the quantum gate: $R_{Z_1}(\theta) = \begin{pmatrix} e^{-k\frac{\theta}{2}} & 0 \\ 0 & e^{k\frac{\theta}{2}} \end{pmatrix}$ is to be applied.

Step 1: Classical Measurement/Computation

Suppose, at an earlier stage of the algorithm, some classical measurements or computations were performed and yielded:

- $A = 5.0$
- $B = 5.0$

Step 2: Calculating the Rotation Angle

Now, attempting to calculate the angle ϕ for the Z-rotation gives:

$$\phi = \frac{C}{A-B} = \frac{\frac{\pi}{2}}{5.0-5.0} = \frac{\frac{\pi}{2}}{5.0-5.0} = \frac{\pi}{2}\left(\frac{1}{0}\right) = \frac{\pi}{2}p$$

This expression results in a division by zero that can be easily handled by the underlying number set upon which the 211-Framework is based.

Step 3: Implication for the Quantum Gate

In the classical quantum computing setting (that is under the 111-Framework where complex numbers are used) the calculation of ϕ results in an undefined value, the classical control system that is supposed to prepare and apply the quantum $R_{Z_1}(\theta)$ gate cannot determine the parameters for the gate. The program would crash or throw an error at Step 2, long before any quantum hardware or simulator attempts to apply the gate.

However, under the 211-Framework, the classical control system would simply pass the value $\frac{\pi}{2}p$ to the quantum system to be handles by the new $R_{Z_1}(\theta)$ gate. This is shown below:

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 September 2025 doi:10.20944/preprints202509.1320.v1

$$R_{Z_1}\left(\frac{\pi}{2}p\right) = e^{-k\frac{\frac{\pi}{2}p}{2}Z_1} = e^{-kp\frac{\pi}{4}Z_1} = e^{i\frac{\pi}{4}Z_1} = I\cos\left(\frac{\pi}{4}\right) + i\sin\left(\frac{\pi}{4}\right)Z_1$$

$$R_{Z_1}\left(\frac{\pi}{2}p\right) = \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}\cos\left(\frac{\pi}{4}\right) + i\sin\left(\frac{\pi}{4}\right)\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$$

$$R_{Z_1}(\theta) = \begin{pmatrix}\cos\left(\frac{\pi}{4}\right) & 0\\ 0 & \cos\left(\frac{\pi}{4}\right)\end{pmatrix} + \begin{pmatrix}i\sin\left(\frac{\pi}{4}\right) & 0\\ 0 & -i\sin\left(\frac{\pi}{4}\right)\end{pmatrix}$$

$$R_{Z_1}(\theta) = \begin{pmatrix}\cos\left(\frac{\pi}{4}\right) + i\sin\left(\frac{\pi}{4}\right) & 0\\ 0 & \cos\left(\frac{\pi}{4}\right) - i\sin\left(\frac{\pi}{4}\right)\end{pmatrix}$$

Recall Eulers formula: $e^{k\theta} = \cos(\theta) + k\sin(\theta)$ hence:

$$R_{Z_1}\left(\frac{\pi}{2}p\right) = \begin{pmatrix}e^{i\frac{\pi}{4}} & 0\\ 0 & e^{-i\frac{\pi}{4}}\end{pmatrix} \tag{21}$$

This rotation is fully achievable under the 211-Framework. This matrix is perfectly well-defined. The classical control system passes these parameters to the quantum hardware (or simulator), and the quantum $R_{Z_1}\left(\frac{\pi}{2}p\right)$ gate is successfully applied to the target qubit. For example, if the qubit is in the state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$):

$$R_{Z_1}\left(\frac{\pi}{2}p\right) \times |+\rangle = R_{Z_1}\left(\frac{\pi}{2}p\right) \times \left[\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}\right] = \frac{1}{\sqrt{2}}\begin{pmatrix}e^{i\frac{\pi}{4}}\\ e^{-i\frac{\pi}{4}}\end{pmatrix}$$

This is a valid quantum state that can easily be represented on the k-Bloch sphere. Proof of this is given in Appendix A6. This example demonstrates that when the classical calculations that determine quantum gate parameters are well-defined, the quantum logic gates operate perfectly normally as unitary transformations.

4. A Note on the 121-Framework and the 112-Framework

For the sake of brevity, a foundation of only one (211-Framework) out of the four mathematical frameworks was established and the utility of the framework demonstrated. The 111-Framework (a complex number mathematical framework) for quantum computing is already well established. The other two frameworks, the 121-Framework and the 112-Framework can be given the same treatment as the 211-Framework from this paper. However, this will not be done here. It suffices to say that these other frameworks are equally good at handling division by zero within quantum computing. A look at these other frameworks will be done in subsequent research papers.

Discussion

Quantum computing, rooted in the mathematical framework of linear algebra over complex numbers, using Hilbert spaces to represent quantum states (qubits) and unitary operations (quantum gates) to manipulate them. Phenomenon like superposition, entanglement, and interference are all precisely described within this mathematical framework. If a different mathematical framework were to be used to create quantum computing, it would be a profound shift providing far-reaching implications.

This research provides the impetus for this shift. 3 mathematical frameworks were developed that can be used for quantum computing. These frameworks are 211-Framework, 121-Framework and 112-Framework. What makes these frameworks fundamentally different from the original 111-Framework (that is, mathematical framework of linear algebra over complex numbers) is that They all incorporate division by zero within their structure in slightly different ways. The frameworks themselves are distinct by the number sets and Pauli matrices that underlie their structure.

 doi:10.20944/preprints202509.1320.v1

Qubits as we know them (superpositions of 0 and 1) are a direct consequence of the current mathematical framework. These new frameworks have the potential to lead to entirely different "quantum bits" with unique properties. It has already been demonstrated in this paper that the fireworks themselves have led to new types of quantum logic gates that operate differently. This means that these new mathematical frameworks have the potential to create entirely new types of quantum algorithms with different strengths and weaknesses to those that already exist. The new framework might unlock computational capabilities that are not possible or even conceivable within the current quantum computing paradigm. Depending on the framework, it might become significantly easier to build robust and scalable quantum computers, or it might introduce even greater engineering challenges.

Other impacts that can be realized from these new frameworks include: (1) entirely new quantum correction algorithms providing a more natural way to tackle optimization problems, machine learning, or complex simulations; (2) new quantum programming computational models; (3) introduce more potent attacks on current cryptographic systems, or conversely, lead to fundamentally new and stronger forms of quantum-resistant cryptography; (4) allow for even more accurate and efficient simulations; and, (5) open up new paradigms for quantum machine learning and AI, potentially leading to breakthroughs in areas like pattern recognition, data analysis, and intelligent systems.

All these possibilities can be explored for each of the new frameworks developed in this paper. In essence, a shift to a different mathematical framework for quantum computing would be a paradigm shift on That has the potential for large scale advancement.

Conclusion

Quantum computing, rooted in the mathematical framework of linear algebra over complex numbers, using Hilbert spaces to represent quantum states (qubits) and unitary operations (quantum gates) to manipulate them. Phenomenon like superposition, entanglement, and interference are all precisely described within this mathematical framework. However, quantum computing is limited by its inability to divide by zero. This curtails the ability of quantum computers to handle singularities and infinities in simulations and solve ill-posed problems. This paper aimed to use semi structured complex numbers to define three distinct number sets that can enable division by zero and simultaneously form four core foundational yet distinct mathematical frameworks for building quantum computers. In the process of doing these three new number sets that are subsets of semi-structured complex numbers were defined; (2) a Generalized Pauli Matrix Sets consisting of 6 Pauli matrices was developed; (3) from these four mathematical frameworks were created and the consistency and full capability of one of the frameworks (221-Framework) was demonstrated. It is expected that these different mathematical frameworks would have a profound shift and provide far-reaching implications for quantum computing especially in the areas of algorithm creation, cryptography and the paradigms for quantum machine learning and AI.

Appendix A

Appendix A1. Major Results from Past Paper on Semi-Structured Complex Numbers

Table A1. Major results from paper [38] for semi-structured complex numbers.

Result 1	Semi-structured complex number set can be defined as follows: *A semi-structured complex number is a three-dimensional number of the general form* $h = x + yi + zp$; *that is, a linear combination of real* (1), *imaginary* (i) *and unstructured* (p) *units whose coefficients* x, y, z *are real numbers.*

<table>
<tr><td></td><td>The number h is called semi-structured complex because it contains a structured complex part $(x + yi)$ and an unstructured part (zp).</td></tr>
<tr><td>Result 2</td><td>The unstructured number p was redefined as:
$$p^n = \frac{\sqrt{2} \times cos\left(\frac{\pi}{2}n - \frac{\pi}{4}\right)}{f^n(1)} \quad (18)$$
where $f^n(c)$ is a composite function such that $f(c) = 1 - c$.
Integer powers of p yield the following cyclic results:
$p^1 = \frac{1}{0}$ $p^2 = -1$ $p^3 = -p$ $p^4 = 1$ $p^5 = \frac{1}{0}$ $p^6 = -1$ $p^7 = -p$ $\cdots$</td></tr>
<tr><td>Result 3</td><td>p does not belong to the set of complex numbers $\mathbb{C}$ (that is, $p \notin \mathbb{C}$), but belongs to a higher order number set $\mathbb{H}$ called the set of semi-structured complex numbers such that the set of complex numbers is a subset of $\mathbb{H}$ (that is, $\mathbb{C} \subset \mathbb{H}$).</td></tr>
<tr><td>Result 4</td><td>The field of semi-structured complex numbers was defined, and proof was given that this field obeys the field axioms. This implies (1) the number set can easily be used in everyday algebraic expressions and can be used to solve algebraic problems, (2) the number set can be used to form more complicated structures such as vector spaces and hence solve more complex problems that may involve "division by zero".</td></tr>
<tr><td>Result 5</td><td>Semi-structured complex number set $\mathbb{H}$ does not form an ordered field. For the objects in a field to have an order, operations such as greater than or less than can be applied to these objects. This is because in an ordered field the square of any non-zero number is greater than 0; this is not the case with semi-structured complex numbers.</td></tr>
<tr><td>Result 6</td><td>Semi-structured complex numbers can be represented by points in a 3-dimensional Euclidean xyz-space. The xyz-space consist of three perpendicular axes: the real x-axis, the imaginary y-axis, and the unstructured z-axis. These axes form three perpendicular planes: the real-imaginary xy-plane, the real-unstructured xz-plane, and the imaginary-unstructured yz-plane.</td></tr>
<tr><td>Result 7</td><td>The unit p was used to find a viable solution to the logarithm of zero. The logarithm of zero was found to be:
$$\log 0 = -p\left(\frac{\pi}{2} + 2k\pi\right) \quad (19)$$
where k is some integer value.</td></tr>
<tr><td>Result 8</td><td>The new definition of p provided an unambiguous understanding that $\frac{0}{0} = n$ simply represents 90° clockwise rotation of the vector np from the positive unstructured z-axis to n on the positive real x-axis along the real-unstructured xz-plane. Note that n is any real number.</td></tr>
<tr><td>Result 9</td><td>Semi-structured complex numbers have both a 3D and 4D representation in the form:
$h = x + yi + zp$ (3D form)
$h = A + Bi + Cp + Dip$ (4D form)
Where: x, y, z, A, B, C, D are real numbered scalars and i, p are semi-structured basis units.</td></tr>
<tr><td>Result 10</td><td>Two new Euler formulas were developed.
<table><tr><th>Plane</th><th>Euler formula</th></tr><tr><td>Real imaginary xy-plane</td><td>$e^{i\theta} = \cos\theta + i\sin\theta$</td></tr></table></td></tr>
</table>

Real unstructured xz-plane	$e^{p\theta} = \cos\theta + p\sin\theta$
Imaginary unstructured yz-plane	$e^{-ip\theta} = \cosh\theta - ip\sinh\theta$

When combined with the original Euler formula describes the relationship between trigonometric, hyperbolic, and exponential functions for the entire semi-structured complex Euclidean xyz-space.

Result 11	Semi-structured complex numbers can be used to resolve singularities that may arise in engineering and science equations (because of division by zero) to develop reasonable conclusions in the absence of experimental data.
Result 12	From Result 10 semi-structured complex numbers can present in four forms as given below:

Semi-structured complex number along	Number
Real-imaginary xy-plane	$h_{xy} = x + iy$
Real-unstructured xz-plane	$h_{xz} = x + pz$
Imaginary-unstructured yz-plane	$h_{yz} = iy + pz$
xyz-space	$h = x + iy + pz$

Result 13	The zeroth root of a number h can be found using the equation $$\sqrt[0]{h} = h^p = e^{p\ln h} = \cos(\ln h) + p\sin(\ln h)$$
Result 14	Since $p^1 = \frac{1}{0}$ this implies that $\frac{1}{p} = 0$ which further implies that $-p = 0$
Result 15	Any real number with the semi-structured unit p attached to it is not a physically measurable quantity. That is, kp where k is a real number is not physically measurable (however, k can be calculated given enough information)
Result 16	If a and b measure different (but quantitatively related) aspects of the same object, where a is physically measurable but b is not, then a and b can be combined into one equation in the form $a + bp$

Appendix A1. Proof that $\boldsymbol{i}$*-Numbers,* $\boldsymbol{p}$*-Numbers,* $\boldsymbol{k}$*-Numbers Obey the Field Axioms*

This section is meant to provide proof that the three subsets (i-numbers, p-numbers, k-numbers) obey the 11 field axioms. In summary the 11 field axioms can be divided into two distinct sets of axioms; the first six axioms address the operation of addition, and the last five axioms address the operation of multiplication.

The proof three subsets (i-numbers, p-numbers, k-numbers) obey the first six axioms is the same and is presented in Table A2. For the last five axioms the i-numbers and p-numbers have the same proof structure and the k-numbers have a different proof structure. These proof structures are presented in Table A3.

Definitions:

To understand these proofs, first, the following definitions are provided:

- In the case of i-numbers h_i, these can be written as $h_i = (a, b)$, where $a, b \in \mathbb{R}$. This is often represented as $a + bi$, where i is the imaginary unit.
- In the case of p-numbers h_p, these can be written as $h_p = (a, b)$, where $a, b \in \mathbb{R}$. This is often represented as $a + bp$, where p is the unstructured unit.

From these definitions' equality, addition, and multiplication can be defined as follows:

 doi:10.20944/preprints202509.1320.v1

Equality:	$(a, b) = (c, d)$ *if and only if* $a = c$ *and* $b = d$
Addition:	$(a, b) + (c, d) = (a + c, b + d)$
Multiplication:	$(a, b) \cdot (c, d) = (ac - bd, ad + bc)$

Additionally,

- In the case of k-numbers h_k, these can be written as $h_k = (a, b)$, where $a, b \in \mathbb{R}$. This is often represented as $a + bk$, where k is the imaginary unstructured unit.

From this definition equality, addition, and multiplication can be defined as follows:

Equality:	$(a, b) = (c, d)$ *if and only if* $a = c$ *and* $b = d$
Addition:	$(a, b) + (c, d) = (a + c, b + d)$
Multiplication:	$(a, b) \cdot (c, d) = (ac + bd, ad + bc)$

Given these definitions the proofs as follows:

Table A2. Proof that all three number sets obey the first six field axioms.

Axioms	Proof
A1: Closure under Addition	For any $(a, b), (c, d) \in \mathbb{C}$, their sum $(a + c, b + d)$ is also an ordered pair of real numbers, since $a + c \in \mathbb{R}$ and $b + d \in \mathbb{R}$. Thus, $(a + c, b + d) \in \mathbb{C}$.
A2: Associativity of Addition	Let $(a, b), (c, d), (e, f) \in \mathbb{C}$. Then $\big((a, b) + (c, d)\big) + (e, f) = (a + c, b + d) + (e, f) = \big((a + c) + e, (b + d) + f\big)(a, b) + \big((c, d) + (e, f)\big) = (a, b) + (c + e, d + f) = \big(a + (c + e), b + (d + f)\big)$. Since addition of real numbers is associative, $(a + c) + e = a + (c + e)$ and $(b + d) + f = b + (d + f)$. Therefore, $((a, b) + (c, d)) + (e, f) = (a, b) + ((c, d) + (e, f))$.
A3: Commutativity of Addition	Let $(a, b), (c, d) \in \mathbb{C}$. $(a, b) + (c, d) = (a + c, b + d)$ $(c, d) + (a, b) = (c + a, d + b)$ Since addition of real numbers is commutative, $a + c = c + a$ and $b + d = d + b$. Therefore, $(a, b) + (c, d) = (c, d) + (a, b)$.
A4: Additive Identity (Zero Element)	The additive identity is $(0,0) \in \mathbb{C}$. For any $(a, b) \in \mathbb{C}$, $(a, b) + (0,0) = (a + 0, b + 0) = (a, b)$.
A5: Additive Inverse	For every $(a, b) \in \mathbb{C}$, its additive inverse is $(-a, -b) \in \mathbb{C}$. $(a, b) + (-a, -b) = (a + (-a), b + (-b)) = (0,0)$.
A6: Closure under Multiplication	For any $(a, b), (c, d) \in \mathbb{C}$, their product $(ac - bd, ad + bc)$ is also an ordered pair of real numbers, since $ac - bd \in \mathbb{R}$ and $ad + bc \in \mathbb{R}$. Thus, $(ac - bd, ad + bc) \in \mathbb{C}$.

Note:

1. Since all eleven field axioms are satisfied, the set of i-numbers (complex numbers $\mathbb{C}$) with the defined addition and multiplication operations forms a field.
2. Please note that the proof that the p-numbers, k-numbers satisfies the first six field axioms are exactly the same as shown in Table A2.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 September 2025 doi:10.20944/preprints202509.1320.v1

Table A3. Proof that the three number sets i-numbers obey the last five field axioms.

Axioms	Proof for i-numbers and p-numbers	Proof for k-numbers
A7: Associativity of Multiplication	Let $(a,b),(c,d),(e,f) \in \mathbb{C}$. $\big((a,b)\cdot(c,d)\big)\cdot(e,f) =$ $(ac-bd, ad+bc)\cdot(e,f) =$ $\begin{pmatrix}(ac-bd)e-(ad+bc)f,\\(ac-bd)f+(ad+bc)e\end{pmatrix} =$ $\begin{pmatrix}ace-bde-adf-bcf,\\acf-bdf+ade+bce\end{pmatrix}$ $(a,b)\cdot\big((c,d)\cdot(e,f)\big) =$ $(a,b)\cdot(ce-df, cf+de) =$ $\begin{pmatrix}a(ce-df)-b(cf+de),\\a(cf+de)+b(ce-df)\end{pmatrix} =$ $\begin{pmatrix}ace-adf-bcf-bde,\\acf+ade+bce-bdf\end{pmatrix}$ Comparing the components, they are equal.	Let $(a,b),(c,d),(e,f) \in \mathbb{K}$. $\big((a,b)\cdot(c,d)\big)\cdot(e,f) =$ $(ac+bd, ad+bc)\cdot(e,f) =$ $\begin{pmatrix}(ac+bd)e+(ad+bc)f,\\(ac+bd)f+(ad+bc)e\end{pmatrix} =$ $\begin{pmatrix}ace+bde+adf+bcf,\\acf+bdf+ade+bce\end{pmatrix}$ $(a,b)\cdot\big((c,d)\cdot(e,f)\big) =$ $(a,b)\cdot(ce+df, cf+de) =$ $\begin{pmatrix}a(ce+df)+b(cf+de),\\a(cf+de)+b(ce+df)\end{pmatrix} =$ $\begin{pmatrix}ace+adf+bcf+bde,\\acf+ade+bce+bdf\end{pmatrix}$ Comparing the components, they are equal.
A8: Commutativity of Multiplication	Let $(a,b),(c,d) \in \mathbb{C}$. $(a,b)\cdot(c,d) =$ $(ac-bd, ad+bc)(c,d)\cdot(a,b) =$ $(ca-db, cb+da)$ Since multiplication and addition of real numbers are commutative, a $c-bd = ca-db$ and $ad+bc = cb+da$. Therefore, $(a,b)\cdot(c,d) = (c,d)\cdot(a,b)$.	Let $(a,b),(c,d) \in \mathbb{K}$. $(a,b)\cdot(c,d) =$ $(ac+bd, ad+bc)(c,d)\cdot(a,b) =$ $(ca+db, cb+da)$ Since multiplication and addition of real numbers are commutative, a $c+bd = ca+db$ and $ad+bc = cb+da$. Therefore, $(a,b)\cdot(c,d) = (c,d)\cdot(a,b)$.
A9: Multiplicative Identity (Unity Element)	The multiplicative identity is $(1,0) \in \mathbb{C}$. For any $(a,b) \in \mathbb{C}$, $(a,b)\cdot(1,0) = (a\cdot 1 - b\cdot 0, a\cdot 0 + b\cdot 1) =$ $(a-0, 0+b) = (a,b)$.	The multiplicative identity is $(1,0) \in \mathbb{K}$. For any $(a,b) \in \mathbb{K}$, $(a,b)\cdot(1,0) = (a\cdot 1 - b\cdot 0, a\cdot 0 + b\cdot 1) =$ $(a-0, 0+b) = (a,b)$.
A10: Multiplicative Inverse	For every non-zero $(a,b) \in \mathbb{C}$ (meaning not both a and b are zero), we need to find (x,y) such that $(a,b)\cdot(x,y) = (1,0)$. Using the multiplication rule: $(ax-by, ay+bx) = (1,0)$. This gives us a system of two linear equations: $ax-by=1, bx+ay=0$	For every non-zero $(a,b) \in \mathbb{K}$ (meaning not both a and b are zero), we need to find (x,y) such that $(a,b)\cdot(x,y) = (1,0)$. Using the multiplication rule: $(ax+by, ay+bx) = (1,0)$. This gives us a system of two linear equations: $ax+by=1, bx+ay=0$

	From the second equation, if $a \neq 0$, then $y = -\frac{bx}{a}$. *Substitute this into the first equation:* $ax - b\left(-\frac{bx}{a}\right) = ax + \frac{b^2x}{a}$ $= \frac{a^2x + b^2x}{a} = 1$ $x(a^2 + b^2) = a$ $x = \frac{a}{a^2 + b^2}$ Now, substitute x back into the expression for y: $y = -\frac{b}{a} \times \frac{a}{a^2 + b^2} = -\frac{b}{a^2 + b^2}$ So, the multiplicative inverse of (a, b) is $\left(\frac{a}{a^2 + b^2}, -\frac{b}{a^2 + b^2}\right)$ Note that $a^2 + b^2 \neq 0$ since $(a, b) \neq (0,0)$. This inverse is well-defined.	From the second equation, if $a \neq 0$, then $y = -\frac{bx}{a}$. *Substitute this into the first equation:* $ax + b\left(-\frac{bx}{a}\right) = ax - \frac{b^2x}{a}$ $= \frac{a^2x - b^2x}{a} = 1$ $x(a^2 - b^2) = a$ $x = \frac{a}{a^2 - b^2}$ Now, substitute x back into the expression for y: $y = -\frac{b}{a} \times \frac{a}{a^2 - b^2} = -\frac{b}{a^2 - b^2}$ So, the multiplicative inverse of (a, b) is $\left(\frac{a}{a^2 - b^2}, -\frac{b}{a^2 - b^2}\right)$ Note that $a^2 - b^2 \neq 0$ since $(a, b) \neq (0,0)$. This inverse is well-defined.
A11: Distributivity of Multiplication over Addition	Let $(a, b), (c, d), (e, f) \in \mathbb{C}$. $(a, b) \cdot \big((c, d) + (e, f)\big) =$ $(a, b) \cdot (c + e, d + f) =$ $\begin{pmatrix} a(c + e) - b(d + f), \\ a(d + f) + b(c + e) \end{pmatrix} =$ $(ac + ae - bd - bf, ad + af + bc + be)$ $(a, b) \cdot (c, d) + (a, b) \cdot (e, f) =$ $(ac - bd, ad + bc) + (ae - bf, af + be) =$ $\begin{pmatrix} (ac - bd) + (ae - bf) \\ (ad + bc) + (af + be) \end{pmatrix}$ $= (ac - bd + ae - bf, ad + bc + af + be)$ Comparing the components, we see they are equal.	Let $(a, b), (c, d), (e, f) \in \mathbb{K}$. $(a, b) \cdot \big((c, d) + (e, f)\big) =$ $(a, b) \cdot (c + e, d + f) =$ $\begin{pmatrix} a(c + e) + b(d + f), \\ a(d + f) + b(c + e) \end{pmatrix} =$ $(ac + ae + bd + bf, ad + af + bc + be)$ $(a, b) \cdot (c, d) + (a, b) \cdot (e, f) =$ $(ac + bd, ad + bc) + (ae - bf, af + be) =$ $\begin{pmatrix} (ac + bd) + (ae + bf) \\ (ad + bc) + (af + be) \end{pmatrix}$ $= (ac + bd + ae + bf, ad + bc + af + be)$ Comparing the components, we see they are equal.

Appendix A2. Sample Proof of the Hermitian and Unitary Nature of Matrix Transformation $\boldsymbol{T_5}$ *and Matrix Transformation* T_7

It would be a bit cumbersome to attempt to do a proof of the Hermitian and unitary nature of all eight matrix transformations in Table 8. Therefore, two matrix transformations were selected at random and the proof of these transformations was given here as a sample of how to prove the Hermitian and unitary nature of the matrix transformations presented in Table 8. The two transformations selected were matrix transformation T_5 and matrix transformation T_7

1. ***Sample Proof of the Hermitian and Unitary nature of matrix transformation*** $\boldsymbol{T_5}$

Consider the matrix

$$V = \begin{bmatrix} v_3 & v_1 + v_2 p \\ v_1 - v_2 p & -v_3 \end{bmatrix} \tag{20}$$

Hermitian Property:

The transpose of this matrix is given below:

$$V^T = \begin{bmatrix} v_3 & v_1 - v_2 p \\ v_1 + v_2 p & -v_3 \end{bmatrix} \tag{21}$$

The complex conjugate of the transpose is given as:

$$V^* = \begin{bmatrix} v_3 & v_1 - v_2 p \\ v_1 - v_2 p & -v_3 \end{bmatrix} \tag{22}$$

Clearly Equation (20) and (22) are equivalent. Thus, the Hermitian property is proven.

Additionally, if the matrices in Equation (20) and (22) are multiplied:

$$VV^* = \begin{bmatrix} v_3 & v_1 + v_2 p \\ v_1 - v_2 p & -v_3 \end{bmatrix} \begin{bmatrix} v_3 & v_1 + v_2 p \\ v_1 - v_2 p & -v_3 \end{bmatrix}$$

$$VV^* = \begin{bmatrix} (v_3)^2 + (v_1 + pv_2)(v_1 - pv_2) & v_3(v_1 + pv_2) - v_3(v_1 + pv_2) \\ v_3(v_1 - pv_2) + v_3(v_1 + pv_2) & (v_1 - pv_2)(v_1 + pv_2) + (v_3)^2 \end{bmatrix}$$

$$VV^* = \begin{bmatrix} (v_3)^2 + (v_1 + pv_2)(v_1 - pv_2) & v_3(v_1 + pv_2) - v_3(v_1 + pv_2) \\ v_3(v_1 - pv_2) - v_3(v_1 - pv_2) & (v_1 - pv_2)(v_1 + pv_2) + (v_3)^2 \end{bmatrix}$$

$$VV^* = \begin{bmatrix} (v_1)^2 - (pv_2)^2 + (v_3)^2 & 0 \\ 0 & (v_1)^2 - (pv_2)^2 + (v_3)^2 \end{bmatrix}$$

Since $(p)^2 = -1$

$$VV^* = \begin{bmatrix} (v_3)^2 + (v_2)^2 + (v_3)^2 & 0 \\ 0 & (v_3)^2 + (v_2)^2 + (v_3)^2 \end{bmatrix}$$

But (v_1, v_2, v_3) is a unit vector. This implies that $(v_3)^2 + (v_2)^2 + (v_3)^2 = 1$. Hence

$$VV^* = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{23}$$

Therefore the unitary property has been proven.

Therefore the matrix $\begin{bmatrix} v_3 & v_1 + pv_2 \\ v_1 - pv_2 & -v_3 \end{bmatrix}$ is a Hermitian unitary matrix.

2. ***Sample Proof of the Hermitian and Unitary nature of matrix transformation*** $\boldsymbol{T_7}$

Additionally, consider the matrix:

$$V = \begin{bmatrix} kv_3 & v_1 + pv_2 \\ v_1 - pv_2 & -kv_3 \end{bmatrix} \tag{24}$$

Hermitian Property:

The transpose of this matrix is given below:

$$V^T = \begin{bmatrix} kv_3 & v_1 - pv_2 \\ v_1 + pv_2 & -kv_3 \end{bmatrix} \tag{25}$$

The complex conjugate of the transpose is given as:

$$V^* = \begin{bmatrix} kv_3 & v_1 - pv_2 \\ v_1 - pv_2 & -kv_3 \end{bmatrix} \tag{26}$$

Clearly Equation (20) and (22) are equivalent. Thus the Hermitian property is proven.

Additionally, if the matrices in Equation (20) and (22) are multiplied:

$$VV^* = \begin{bmatrix} kv_3 & v_1 + pv_2 \\ v_1 - pv_2 & -kv_3 \end{bmatrix} \begin{bmatrix} kv_3 & v_1 + pv_2 \\ v_1 - pv_2 & -kv_3 \end{bmatrix}$$

$$VV^* = \begin{bmatrix} (kv_3)^2 + (v_1 + pv_2)(v_1 - pv_2) & kv_3(v_1 + pv_2) - kv_3(v_1 + pv_2) \\ kv_3(v_1 - pv_2) + kv_3(v_1 + pv_2) & (v_1 - pv_2)(v_1 + pv_2) + (kv_3)^2 \end{bmatrix}$$

$$VV^* = \begin{bmatrix} (kv_3)^2 + (v_1 + pv_2)(v_1 - pv_2) & kv_3(v_1 + pv_2) - kv_3(v_1 + pv_2) \\ kv_3(v_1 - pv_2) - kv_3(v_1 - pv_2) & (v_1 - pv_2)(v_1 + pv_2) + (kv_3)^2 \end{bmatrix}$$

$$VV^* = \begin{bmatrix} (v_1)^2 - (pv_2)^2 + (kv_3)^2 & 0 \\ 0 & (v_1)^2 - (pv_2)^2 + (kv_3)^2 \end{bmatrix}$$

Since $(k)^2 = 1$

$$VV^* = \begin{bmatrix} (v_3)^2 + (v_2)^2 + (v_3)^2 & 0 \\ 0 & (v_3)^2 + (v_2)^2 + (v_3)^2 \end{bmatrix}$$

But (v_1, v_2, v_3) is a unit vector. This implies that $(v_3)^2 + (v_2)^2 + (v_3)^2 = 1$. Hence

$$VV^* = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{27}$$

Therefore, the unitary property has been proven.

Proof for the other six transformation matrices in Table 8 follow the same basic steps.

Appendix A3. Commutator Identities Under the 211-Framework

Consider the commutation relation $[A, B] = AB - BA$. This relation can be applied to pairs of matrices from the set $\{X_2, Y_1, Z_1\}$.

First identity

$$[X_2, Y_1] = X_2Y_1 - Y_1X_2$$

$$[X_2, Y_1] = \begin{pmatrix} 0 & k \\ k & 0 \end{pmatrix} \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix} - \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix} \begin{pmatrix} 0 & k \\ k & 0 \end{pmatrix}$$

$$[X_2, Y_1] = \begin{pmatrix} -ik & 0 \\ 0 & ik \end{pmatrix} - \begin{pmatrix} ik & 0 \\ 0 & -ik \end{pmatrix}$$

$$[X_2, Y_1] = \begin{pmatrix} -2ik & 0 \\ 0 & 2ik \end{pmatrix}$$

$$[X_2, Y_1] = -2ik \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = 2p \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad \text{since} \quad -2ik = -2i(ip) = 2p$$

$$[X_2, Y_1] = 2pZ_1$$

Second identity

$$[Y_1, Z_1] = Y_1Z_1 - Z_1Y_1$$

$$[Y_1, Z_1] = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}$$

$$[Y_1, Z_1] = \begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix} - \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix}$$

$$[Y_1, Z_1] = \begin{pmatrix} 0 & -2i \\ -2i & 0 \end{pmatrix} = \begin{pmatrix} 0 & -2ik^2 \\ -2ik^2 & 0 \end{pmatrix} \qquad \text{since } k^2 = 1$$

$$[Y_1, Z_1] = -2ik \begin{pmatrix} 0 & k \\ k & 0 \end{pmatrix} = 2p \begin{pmatrix} 0 & k \\ k & 0 \end{pmatrix} \qquad \text{since } -2ik = -2i(ip) = 2p$$

$$[Y_1, Z_1] = 2pX_2$$

Third identity

$$[Z_1, X_2] = Z_1X_2 - X_2Z_1$$

$$[Z_1, X_2] = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 0 & k \\ k & 0 \end{pmatrix} - \begin{pmatrix} 0 & k \\ k & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$[Z_1, X_2] = \begin{pmatrix} 0 & k \\ -k & 0 \end{pmatrix} - \begin{pmatrix} 0 & -k \\ k & 0 \end{pmatrix}$$

$$[Z_1, X_2] = \begin{pmatrix} 0 & 2k \\ -2k & 0 \end{pmatrix} = \begin{pmatrix} 0 & 2ip \\ -2ip & 0 \end{pmatrix} = 2p \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}$$

$$[Z_1, X_2] = 2pY_1$$

Appendix A4. Proving the Universality of the Set $\{\boldsymbol{H}, \boldsymbol{T_k}\}$ *Under the 211-Framework*

To prove that the set $\{H, T_k\}$ we use the Solovay Kitaev Theorem. The theorem states that: "*if a set of single-qubit gates can generate a dense subgroup of SU(2), then any desired single-qubit gate can be approximated to an arbitrary precision* ϵ *using a sequence of gates from this finite set*". This finite set of gates is then called the universal set. The key to proving that $\{H, T_k\}$ forms a universal set is to simply prove that they form a "dense subgroup" of SU(2).

The key principle for denseness is that A finite set of rotations on a sphere generates a dense set of rotations if and only if the group they generate contains rotations about two axes that are not parallel.

We simply need to prove that the set $\{H, T_k\}$ can generate rotations about two non-parallel axes.

PART 1: Proving that the $\boldsymbol{T_k}$ is fundamentally a Z-axis rotation.

1. The T-gate is defined as:

$$T_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{k\frac{\pi}{4}} \end{pmatrix} \tag{28}$$

2. A rotation around the Z-axis by an angle θ is given by:

$$R_{Z_1}(\theta) = e^{-k\frac{\theta}{2}Z_1} = I\cos\left(\frac{\theta}{2}\right) - k\sin\left(\frac{\theta}{2}\right)Z_1$$

$$R_{Z_1}(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\cos\left(\frac{\theta}{2}\right) - k\sin\left(\frac{\theta}{2}\right)\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$R_{Z_1}(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & 0 \\ 0 & \cos\left(\frac{\theta}{2}\right) \end{pmatrix} - \begin{pmatrix} k\sin\left(\frac{\theta}{2}\right) & 0 \\ 0 & -k\sin\left(\frac{\theta}{2}\right) \end{pmatrix}$$

$$R_{Z_1}(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) - k\sin\left(\frac{\theta}{2}\right) & 0 \\ 0 & \cos\left(\frac{\theta}{2}\right) + k\sin\left(\frac{\theta}{2}\right) \end{pmatrix}$$

Recall Eulers formula: $e^{k\theta} = \cos(\theta) + k\sin(\theta)$ hence: (29)

$$R_{Z_1}(\theta) = \begin{pmatrix} e^{-k\frac{\theta}{2}} & 0 \\ 0 & e^{k\frac{\theta}{2}} \end{pmatrix}$$

3. Comparing T-gate with $R_{Z_1}\left(\frac{\theta}{2}\right)$ and the concept of global phase equivalence

It is necessary to find an angle θ such that T is equivalent to $R_{Z_1}(\theta)$. Recall in quantum mechanics, two gates U_1 and U_2 are considered equivalent if they differ only by a global phase factor, i.e., $U_1 = e^{k\phi}U_2$ for some real ϕ. This is because a global phase does not affect the physical probabilities of measurement outcomes. Hence, we need to solve the equation

$$T_k = e^{k\phi} R_{Z_1}\left(\frac{\theta}{2}\right)$$

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{k\frac{\pi}{4}} \end{pmatrix} = e^{k\phi} \begin{pmatrix} e^{-k\frac{\theta}{2}} & 0 \\ 0 & e^{k\frac{\theta}{2}} \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{k\frac{\pi}{4}} \end{pmatrix} = \begin{pmatrix} e^{k\left(-\frac{\theta}{2}+\phi\right)} & 0 \\ 0 & e^{k\left(\frac{\theta}{2}+\phi\right)} \end{pmatrix}$$

Recall that $e^{k(0)} = 1$. Hence:

$$\begin{pmatrix} e^{k(0)} & 0 \\ 0 & e^{k\frac{\pi}{4}} \end{pmatrix} = \begin{pmatrix} e^{k\left(-\frac{\theta}{2}+\phi\right)} & 0 \\ 0 & e^{k\left(\frac{\theta}{2}+\phi\right)} \end{pmatrix}$$

Now this implies that the following simultaneous equation needs to be solved:

$$-\frac{\theta}{2} + \phi = 0$$

$$\frac{\theta}{2} + \phi = \frac{\pi}{4}$$

The solution to these pair of equations is:

$$\frac{\theta}{2} = \phi = \frac{\pi}{8}$$

Hence:

$$T_k = e^{k\frac{\pi}{8}} R_{Z_1}\left(\frac{\pi}{4}\right) \qquad (30)$$

This shows that the T-gate is equivalent to a Z-axis rotation by an angle of $\theta = \frac{\pi}{4}$, up to a global phase of $e^{k\frac{\pi}{8}}$. This means the T-gate itself is fundamentally a Z-axis rotation.

PART 2: The non-axial nature of the Hadamard gate

The Hadamard gate is a "non-axial" rotation. It maps the Pauli axes as follows:

- $HX_2H = Z_1$
- $HY_1H = -Y_1$
- $HZ_1H = X_2$

This property is important. If we can perform rotations around the Z-axis (using T), and to "switch" the axes (using H), then we can effectively perform rotations around other axes.

PART 3: Summary

1. The T-gate provides rotations around the Z-axis.
2. The Hadamard gate allows us to transform rotations around the Z-axis into rotations around the X-axis (and vice-versa, and Y-axis rotations into negative Y-axis rotations).
3. Having rotations around two non-parallel axes (e.g., X and Z) with "incommensurate" angles (angles that are irrational multiples of 2π) is sufficient to densely cover the $SU(2)$ group. The $\frac{\pi}{4}$ angle from the T-gate provides this irrationality, as $\frac{\pi}{4}$ is not a rational multiple of 2π.

Therefore, the set $\{H, T_k,\}$ (and its inverses, which are implicitly included as a sequence of gates can always be inverted) is a universal gate set for $SU(2)$ because the group it generates is dense in $SU(2)$.

As similar proof can be used to find the Universal Gate Sets for each quantum computing mathematical framework. A summary of the results of such proofs is provided in Table A4.

Table 19. Universal Gate Sets for each quantum computing mathematical framework

Name of Framework	Universal Gate Set	Matrix representation of gates within Universal Gate Set
111–Framework	$\{H,T\}$	$H=\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$ $T=\begin{pmatrix}1 & 0\\ 0 & e^{i\frac{\pi}{4}}\end{pmatrix}$
211–Framework	$\{H, T_k\}$	$H=\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$ $T_k=\begin{pmatrix}1 & 0\\ 0 & e^{k\frac{\pi}{4}}\end{pmatrix}$
121–Framework	$\{H,T_p\}$	$H=\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$ $T_p=\begin{pmatrix}1 & 0\\ 0 & e^{p\frac{\pi}{4}}\end{pmatrix}$
112–Framework	$\{H, T_{-k}\}$	$H=\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$ $T_{-k}=\begin{pmatrix}1 & 0\\ 0 & e^{-k\frac{\pi}{4}}\end{pmatrix}$

Note:
1. H is called the Hadamar gate, T is called the T-gate.
2. For each gate set the inverses of the gates in the set are implicitly included as a sequence of gates can always be inverted

Appendix A5. Representing a Rotation on the k-Bloch Sphere

Suppose the rotation $R_{Z_1}\left(\frac{\pi}{2}p\right)\times|+\rangle=\frac{1}{\sqrt{2}}\begin{pmatrix}e^{i\frac{\pi}{4}}\\ e^{-i\frac{\pi}{4}}\end{pmatrix}$ needs to be represented on the k-Bloch sphere.

This can be done as follows:

$$\frac{1}{\sqrt{2}}\begin{pmatrix}e^{i\frac{\pi}{4}}\\ e^{-i\frac{\pi}{4}}\end{pmatrix}=\frac{1}{\sqrt{2}}e^{i\frac{\pi}{4}}\begin{pmatrix}1\\0\end{pmatrix}+\frac{1}{\sqrt{2}}e^{-i\frac{\pi}{4}}\begin{pmatrix}0\\1\end{pmatrix}=\frac{1}{\sqrt{2}}e^{i\frac{\pi}{4}}|0>++\frac{1}{\sqrt{2}}e^{-i\frac{\pi}{4}}|1>$$

Removing the global phase $e^{i\frac{\pi}{4}}$ (recall only relative phase between the qubits are important) gives:

$$\frac{1}{\sqrt{2}}\begin{pmatrix}e^{i\frac{\pi}{4}}\\ e^{-i\frac{\pi}{4}}\end{pmatrix}\equiv\frac{1}{\sqrt{2}}|0>++\frac{1}{\sqrt{2}}e^{-i\frac{\pi}{2}}|1>=\frac{1}{\sqrt{2}}|0>++\frac{1}{\sqrt{2}}e^{-i\frac{\pi}{2}}|1>\ =$$

To map this on to the Block Sphere we need to use the equation: $\cos(\frac{\theta}{2})|0>++e^{i\delta}\ \sin(\frac{\theta}{2})|1>$. That is:

$$\frac{1}{\sqrt{2}}|0>++\frac{1}{\sqrt{2}}e^{-i\frac{\pi}{2}}|1>=\cos(\frac{\theta}{2})|0>++e^{i\delta}\ \sin(\frac{\theta}{2})|1>$$

Comparing the left- and right-hand side of the above equation yields the following:

$$\cos(\frac{\theta}{2})=\sin(\frac{\theta}{2})=\frac{1}{\sqrt{2}}\qquad \text{this implies that }\ \theta=\frac{\pi}{2}$$

Additionally,

$$e^{-i\frac{\pi}{2}} = e^{i\delta} \qquad \text{this implies that } \delta = \frac{\pi}{2}$$

The spherical coordinates that map the above results to the k-Bloch sphere are:

$$x = sin\,\theta\cos\delta = sin\left(\frac{\pi}{2}\right)\cos\left(-\frac{\pi}{2}\right) = 0$$

$$y = sin\,\theta\sin\delta = sin\left(\frac{\pi}{2}\right)\sin\left(-\frac{\pi}{2}\right) = -1$$

$$z = cos\,\theta = cos\left(\frac{\pi}{2}\right) = 0$$

Hence the vector on the Bloch sphere becomes $(0,-1,0)$. A visual of the results are given in

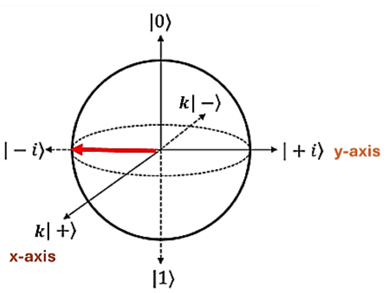

Figure .

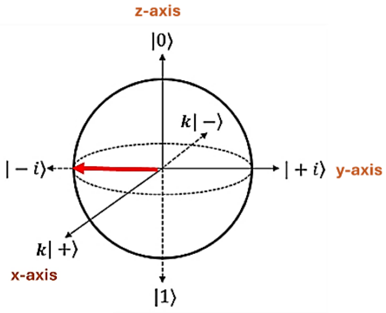

Figure A1. Bloch sphere vector (red arrow) describing the division by zero rotation $\frac{1}{\sqrt{2}}\begin{pmatrix} e^{i\frac{\pi}{4}} \\ e^{-i\frac{\pi}{4}} \end{pmatrix}$ in the 211-Framework.

References

1. Rao, K.J., Rajasekhar, Y., Vijayalakshmi Y. and Malathi, N.V., "Mathematical foundations and algorithms for quantum computing," International Journal of Statistics and Applied Mathematics, vol. 10, no. 3, pp. 97-102, 2025.
2. Naik L.S., "The Role of Linear Algebra in Quantum Computing," IJRAR, vol. 8, no. 1, 2021.
3. Hrdina, J., Hildenbrand, D., Návrat, A. et al., "Quantum Register Algebra: the mathematical language for quantum computing," Quantum Inf Process, vol. 22, pp. https://doi.org/10.1007/s11128-023-04086-y, 2023.
4. Tafadzwa C.E., "Division by zero in real and complex number arithmetic and some of its implications," African Journal of Mathematics and Computer Science Research, vol. 15, no. 2, pp. 14-25, 2022.
5. Mohan, V. , "Black hole singularities from holographic complexity," Journal of High Energy Physics , vol. 2025, p. https://doi.org/10.1007/JHEP07(2025)275, 2025.

6. Frellesvig, H., Morales, R., Pögel, S., Weinzierl, S., & Wilhelm, M., "Calabi-Yau Feynman integrals in gravity ε-factorized form for apparent singularities," Journal of High Energy Physics, vol. 2025, p. https://doi.org/10.1007/JHEP02(2025)209, 2025.
7. Towne B.P., "Quantum Collapse of Indeterminate States: An Alternative Framework for Division by Zero," Yiwu Industrial Commercial College, Yiwu, China∗, 3 October 2024. [Online]. Available: https://vixra.org/pdf/2410.0028v1.pdf. [Accessed 22 August 2025].
8. Di Meglio, A., Jansen, K., Tavernelli, I., Alexandrou, C., Arunachalam, S., Bauer, C. W., Zhang, J. et al., " (2024). Quantum computing for high-energy physics: State of the art and challenges," Prx quantum, vol. 5, no. 3, p. 037001, 2024.
9. Grimm, T.W., van Vliet, M. , "On the complexity of quantum field theory," Journal of High Energy Physics, vol. 2025, p. https://doi.org/10.1007/JHEP06(2025)215, 2025.
10. Meyenburg, T., "A Novel Algebraic Framework for Division by Zero Using Boolean Operations," International Journal of Mathematics Trends and Technology-IJMTT, vol. 71, 2025.
11. Kulkarni, S. P., & Bethel, E. W., "From Bits to Qubits: Challenges in Classical-Quantum Integration," IEEE 31st International Conference on High Performance Computing, Data, and Analytics (HiPC), pp. 166-176.
12. Bharos, N., Markovich, L., & Borregaard, J., "Efficient high-dimensional entangled state analyzer with linear optics.," Quantum, vol. 9, p. 1711.
13. Bärlin, J., "Formation of singularities in solutions to nonlinear hyperbolic systems with general sources.," Nonlinear Analysis: Real World Applications, vol. 73, p. 103901, 2023.
14. An, D., Liu, JP., Wang, D. et al. , "Quantum Differential Equation Solvers: Limitations and Fast-Forwarding," Communications in Mathematical Physics , vol. 406, pp. https://doi.org/10.1007/s00220-025-05358-7, 2025.
15. Shvartsman, I., "Linear Programming Approach to Optimal Control Problems with Unbounded State Constraint," Journal of Optimization Theory and Applications, vol. 204, no. 2, p. 18, 2025.
16. Cai, S., Li, X., "Optimality and scalarization of robust approximate solutions for semi-infinite vector equilibrium problems," Journal of Inequalities and Applications, vol. 67, no. 2025, pp. https://doi.org/10.1186/s13660-025-03315-5, 2025.
17. P. Jean Paul and S. Wahid, "Applications of Semi-Structured Complex Numbers in Science and Engineering.," Preprints. , p. https://doi.org/10.20944/preprints202303.0368.v1, 2023.
18. Peter, J. P., & Shanaz, W., "Reformulating and Strengthening the Theoretical Foundations of Semi-structured Complex Numbers," World Scientific News, vol. 169, pp. 152-182, 2022.
19. Paul, P. J., & Wahid, S., "Unstructured and Semi-structured Complex Numbers: A Solution to Division by Zero," Pure and Applied Mathematics Journal, vol. 10, no. 2, p. 49, 2021.
20. Jean-Paul, P., & Wahid, S. , "Applications of Semi-Structured Complex Numbers in Science and Engineering," ResearchGate, 2023.
21. Koska, O., Baboulin, M., & Gazda, A., "A tree-approach Pauli decomposition algorithm with application to quantum computing," ISC High Performance 2024 Research Paper Proceedings (39th International Conference) , pp. 1-11, 2024.
22. Vidal Romero, S., & Santos-Suárez, J., "PauliComposer: compute tensor products of Pauli matrices efficiently.," Quantum Information Processing, vol. 22, no. 12, p. 449, 2023.
23. Beyer, M., & Paul, W., "Stern–Gerlach, EPRB and Bell Inequalities: An Analysis Using the Quantum Hamilton Equations of Stochastic Mechanics," Foundations of Physics, vol. 54, no. 2, p. 20, 2024.

 doi:10.20944/preprints202509.1320.v1

24. Tah, R., "Calculating the Pauli Matrix equivalent for Spin-1 Particles and further implementing it to calculate the Unitary Operators of the Harmonic Oscillator involving a Spin-1 System.," hal-02909703f, 2020.
25. Liebert, J., Lemke, Y., Altunbulak, M., Maciazek, T., Ochsenfeld, C., & Schilling, C., "Toolbox of spin-adapted generalized Pauli constraints," Physical Review Research, vol. 7, no. 2, p. 023247, 2025.
26. Sirengo C.W., Marani V., Matuya J., "Mathematical Properties and Characteristics of Pauli Unitary Operators in Quantum Information Theory," IRE Journals, vol. 8, no. 2, 2024.
27. van Melkebeek D., "Lecture 6: Single-Qubit and Two-Party Systems," 2 October 2022. [Online]. Available: https://www.youtube.com/watch?v=FyYfDWD8HhM. [Accessed 22 August 2025].
28. Smith, I. D., Cautrès, M., Stephen, D. T., & Poulsen Nautrup, H., "Optimally Generating su (2 N) Using Pauli Strings," Physical Review Letters, vol. 134, no. 20, p. 200601, 2025.
29. Wong, H.Y., "Bloch Sphere, Quantum Gates, and Pauli Matrices," in In: Quantum Computing Architecture and Hardware for Engineers, Springer, Cham, 2025, pp. https://doi.org/10.1007/978-3-031-78219-0_5.
30. "Introduction to Quantum Information Science – The Bloch Sphere," [Online]. Available: https://qubit.guide/2.10-the-bloch-sphere. [Accessed 22 August 2025].
31. "NE 112 Linear algebra for nanotechnology engineering," University of Waterloo, [Online]. Available: https://ece.uwaterloo.ca/~ne112/Lecture_materials/pdfs/1.6.1%20Field%20axioms.pdf. [Accessed 22 August 2025].
32. Mathematics Stack Exchange, "Axioms of Field," Stack Exchange, [Online]. Available: https://math.stackexchange.com/questions/4898251/axioms-of-field. [Accessed 22 August 2025].
33. Rossi Z. M. , "A Solovay-Kitaev theorem for quantum signal processing," Cornell University, 8 May 2025. [Online]. Available: https://arxiv.org/abs/2505.05468. [Accessed 22 August 2025].
34. De Silva K., Mahasinghe A. & Gunasekara P., "Some remarks on the Solovay–Kitaev approximations in a C*–algebra setting," Palestine Journal of Mathematics, vol. 12, no. 2, pp. 25-29, 2023.
35. "Universal set of quantum gates," ICTP-SAIFR, 2022. [Online]. Available: https://www.ictp-saifr.org/wp-content/uploads/2022/11/ICTP_SAIFR_D1-L2.pdf. [Accessed 22 August 2025].
36. "Solovay–Kitaev Theorem," Wikipedia Contributors, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Solovay%E2%80%93Kitaev_theorem. [Accessed 22 August 2025].
37. Long, J., Zhong, J., & Meng, L. , "The Construction of a Universal Quantum Gate Set for the SU(2)k Anyon Models via GA-Enhanced SK Algorithm.," arXiv., 2025.
38. P. Jean-Paul and S. Wahid, "Reformulating and Strengthening the Theoretical Foundations of Semi-structured Complex Numbers," International Journal of Applied Physics and Mathematics, vol. 12, pp. 34-58, 2022.