

Visualizing Quantum Circuits: State Vector Difference Highlighting and the Half-Matrix

Michael J. McGuffin and Jean-Marc Robert

Abstract—Existing graphical user interfaces for circuit simulators often show small visual summaries of the reduced state of each qubit, showing the probability, phase, purity, and/or Bloch sphere coordinates associated with each qubit. These necessarily provide an incomplete picture of the quantum state of the qubits, and can sometimes be confusing for students or newcomers to quantum computing. We contribute two novel visual approaches to provide more complete information about small circuits. First, to complement information about each qubit, we show the complete state vector, and illustrate the way that amplitudes change from layer-to-layer under the effect of different gates, by using a small set of colors, arrows, and symbols. We call this “state vector difference highlighting”, and show how it elucidates the effect of Hadamard, X, Y, Z, S, T, Phase, and SWAP gates, where each gate may have an arbitrary combination of control and anticontrol qubits. Second, we display pairwise information about qubits (such as concurrence and correlation) in a triangular “half-matrix” visualization. Our open source software implementation, called MuqcsCraft, is available as a live online demonstration that runs in a web browser without installing any additional software, allowing a user to define a circuit through drag-and-drop actions, and then simulate and visualize it.

Index Terms—Quantum circuit, quantum algorithm, graphical user interface, GUI, state vector simulation, entanglement.

1 INTRODUCTION

GRAPHICAL simulators for quantum circuits often show a circuit diagram with small visual summaries of the reduced state (also called local state) of each qubit. These visual summaries might show the measurement probability for the $|1\rangle$ state, the qubit’s phase or coordinates in the Bloch sphere, and the purity in the reduced state¹ (Figure 1). Although this information is useful, it can also be limiting or confusing for beginners: for example, just showing information about single-qubit reduced states can make it difficult to know which qubits are correlated or entangled, and a qubit’s phase does not always change as might be expected in response to a gate. More fundamentally, in a circuit on n qubits, visualizing the reduced state (or any fixed amount of information) for each qubit only shows $O(n)$ information, which *cannot* reveal a complete picture of the circuit’s quantum state, since the state vector comprises $O(2^n)$ degrees of freedom.

As an alternative approach, we propose visualizing all 2^n complex amplitudes of the entire state vector, as they evolve layer-by-layer. This is only feasible for small circuits (roughly $n \leq 8$, or 256 amplitudes), however, circuits of this size are useful for educational purposes and for developing intuition. We show that a simple set of *visual primitives* (colors, arrows, and symbols) is sufficient to visually augment the state vector to elucidate the effect of a core subset of gates (Figure 2), namely Hadamard, X, Y, Z, $S = \sqrt{Z}$,

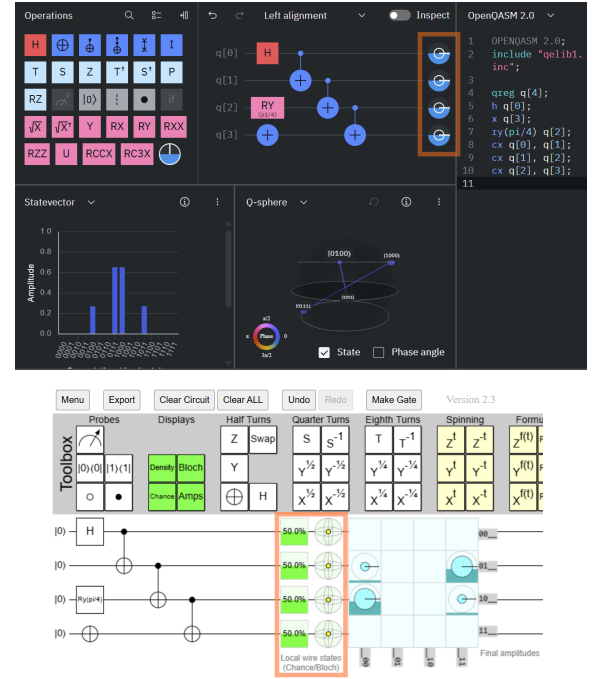


Fig. 1. IBM Quantum Composer (top) [2] and Quirk (bottom) [3] showing the same circuit. In both, the reduced state of the qubits is shown to the right of the circuit diagram, in the area enclosed by the orange rectangle. Inside this orange rectangle, IBM’s “phase disks” show the phase (white tick mark), probability (blue area), and purity (the diameter of the white circle is half of the disk, indicating a purity of 0.5), whereas Quirk’s widgets show the probability (green area) and Bloch sphere coordinates (a yellow dot in the center of each sphere, because the reduced states are maximally mixed, meaning a purity of 0.5).

• M. McGuffin and J.-M. Robert are with École de technologie supérieure, Montreal, Canada.

1. The reduced state of a qubit is obtained by tracing out all other qubits from the density matrix. Sections 6.3 and 6.5 in [1] explain how to compute these and related statistics.

$T = \sqrt[4]{Z}$, Z^k , Phase(θ), GlobalPhase(θ), and SWAP gates²,

2. Phase(θ) is equivalent to $Z^{\theta/\pi}$. GlobalPhase is abbreviated as GP in the screenshots of our software. A complete list of matrix definitions of the gates is available at <https://github.com/MJMcGuffin/muqcs.js?tab=readme-ov-file#matrix-definitions>

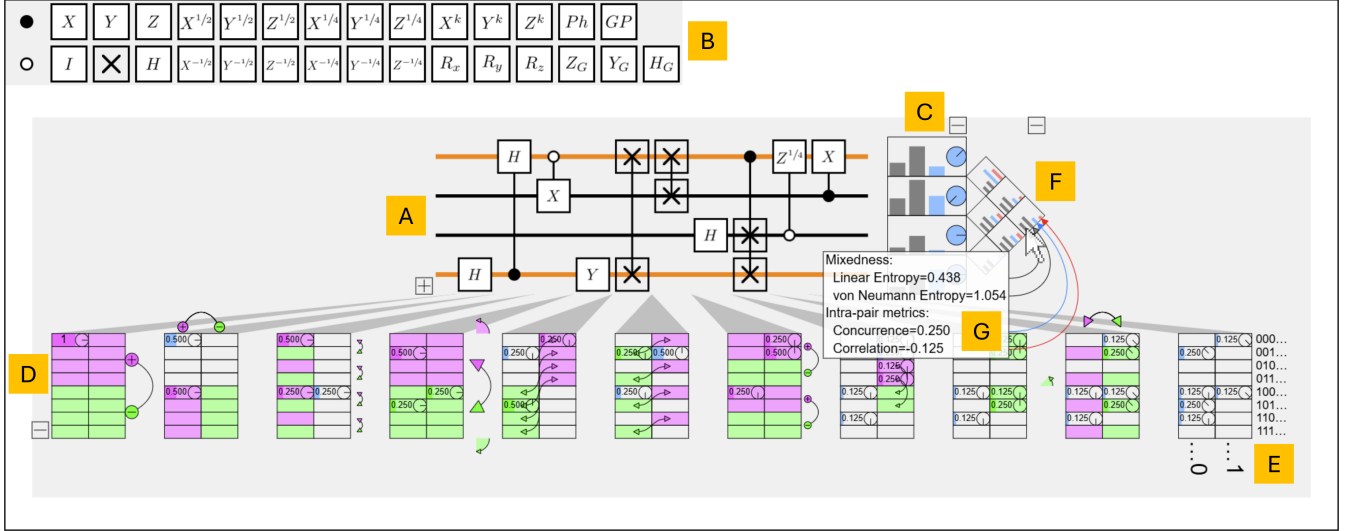


Fig. 2. MuqcsCraft’s visual interface. A: The circuit diagram. Black and white dots correspond to control and anticontrol qubits. B: A toolbar from which gates can be dragged onto the circuit diagram. C: The reduced state of each qubit at the output of the circuit: linear entropy and von Neumann entropy (shown with grey bars), probability of measuring 1 (blue bar), phase (blue disc). D: The state vector, layer-by-layer, shown below the circuit diagram. For each layer of the circuit, the 16 amplitudes of the state vector are wrapped into 8×2 cells. Difference highlighting indicates how the state vector changes under the effect of each gate, using the purple and green colors, arrows, and symbols. E: A key to the bitstrings for the base states. For example, the lower left cell in each 8×2 state vector is 1110, where the left-most bit corresponds to the bottom wire in the circuit. F: The triangular half-matrix shows one cell for each pair of qubits. For example, the mouse cursor is over the right-most cell, corresponding to the top and bottom wires which are highlighted in orange. G: A tooltip with curved callout arrows gives details about the contents of the cell under the cursor.

where each gate may have any combination of control and anticontrol qubits. We call this use of colors, arrows, and symbols *state vector difference highlighting*, because of the way they illustrate the evolution of the state vector (Figure 2 D). Other gates not in the core subset (such as X^k and Y^k) can be replaced with equivalent sequences of gates from the subset, enabling the effect of all single-qubit gates to be visualized with difference highlighting, establishing a kind of *visual universality* which we define later. Thus, a small set of visual primitives makes quantum state dynamics more comprehensible.

We also propose a second novel visualization called the *half-matrix* which shows information about every pair of qubits (Figure 2 F). Our implementation shows the entropy, correlation, and concurrence [4] for each pair of qubits, and the half-matrix can easily show other kinds of pairwise information.

We also present examples demonstrating how state vector difference highlighting can make it more straightforward to design a circuit, and how the half-matrix makes it easier to identify relationships between pairs of qubits that are partially or fully entangled. These tasks are much easier to perform with our techniques than with the status quo visualizations in existing software (Figure 1).

Our software implementation, MuqcsCraft [5], allows the user to define a circuit with drag-and-drop actions. Interactive coordinated highlighting (where the user hovers the cursor over one element, causing corresponding elements elsewhere to highlight) and tooltips with curved callout arrows (Figure 2 G) make the user interface more self-explanatory. MuqcsCraft is open-source and web-based, available to use at <https://mjmcguffin.github.io/MuqcsCraft/> without installing any software. A companion

video³ demonstrates features of the software.

2 PROBLEMS WITH CURRENT APPROACHES

The fact that every unitary gate causes a rotation (and/or a change in global phase) of the state vector in Hilbert space⁴, and that every single-qubit gate causes a rotation in the Bloch sphere (Figure 4 D), suggests an appealingly simple mental model. Beginners might think that, to change the phase of a qubit, they merely need to choose a gate with an appropriate rotation angle. We might therefore hope that the visualizations of qubit reduced states would guide a user in choosing appropriate gates to achieve various effects. Indeed, in simple cases (Figure 3), the effect of certain gates on qubit phase is straightforward.

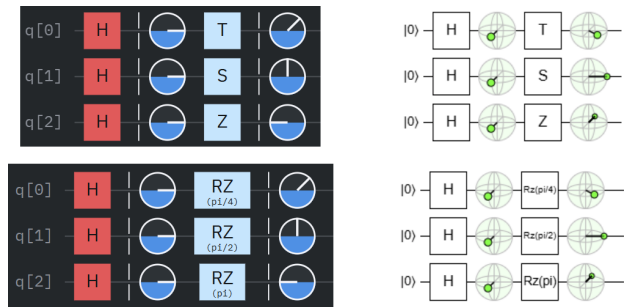


Fig. 3. Visual feedback in IBM Quantum Composer (left) and Quirk (right), when gates have an easy-to-understand effect on the phase, rotating the phase by $\pi/4$, $\pi/2$, and π radians.

3. https://youtu.be/BCunU_gCXT4

4. Any n -dimensional unitary operator can be expressed as $e^{i\theta/n}R$ where R is a unitary rotation matrix with $\det R = 1$.

However, Figure 4 shows how certain cases could be sources of confusion for a student or beginner. First, rotation angles are reduced if there is a control qubit involved. Second, rotations around an axis other than z are not visible in IBM's "phase disks", and can be difficult to see in Quirk's Bloch spheres (for example, how easily can we see that the phase in Figure 4 E is actually $-\pi/2$?). Third, single-qubit gates have no effect on Bloch sphere coordinates if the reduced state is maximally mixed, which is not clearly shown in IBM's user interface. Fourth, phase kickback is another potential source of confusion, complicating the mental model of how gates affect qubits.

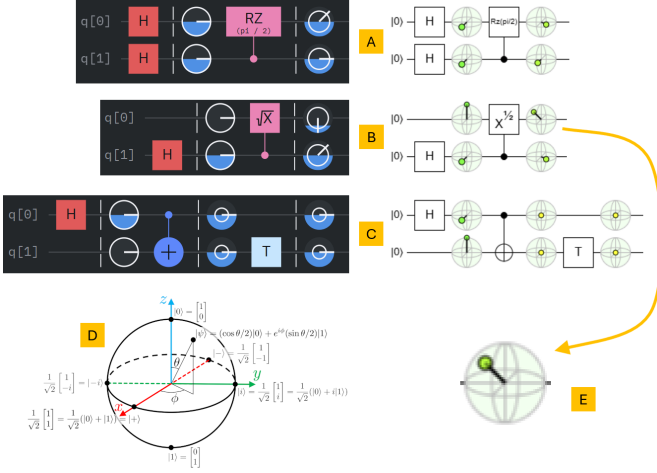


Fig. 4. Examples where gates have a complicated effect on qubit phase, showing the visual feedback of IBM (left) and Quirk (right). A: the RZ gate has an angle of $\pi/2$, but the phase changes by $\pi/4$, because of the control qubit. B: a $\sqrt{X}$ gate normally rotates by $\pi/2$ around the x axis, and in the circuit, we do see a change of $\pi/2$ in $q[0]$'s phase, but in the negative direction. $q[1]$'s phase also changes, because of phase kickback. C: a T gate normally rotates the phase by $\pi/4$, but in this case, we see no change in the phase. Each of A, B, and C could be a source of confusion for a student or beginner. D: the Bloch sphere, where ϕ is the phase, and axes are oriented the same way as in Quirk's visual feedback. E: enlarged view of Quirk's visual feedback in B.

Even for researchers with some experience, it might sometimes be difficult to determine which gate to use for a given purpose. In the next section, we propose showing the state vector, and show how it can be visualized with *difference highlighting*, to make clearer how and why it changes from one layer of the circuit to the next. We have found such visual feedback more helpful for designing circuits than showing qubit reduced states.

Single qubit reduced states can also make it difficult to understand a circuit's operation. For example, Figure 1 shows a circuit similar to one that would generate a GHZ state on four qubits, except that the bottom qubit (which is left-most in bitstrings) is inverted, and the $RY(\pi/4)$ gate perturbs things. The state vector visualizations in both user interfaces in Figure 1 indicate that the base states 0111 and 1000 are more likely than 0100 or 1011, with all other base states having probability zero. This implies that (1) the two top qubits (which are right-most in the bit strings) are perfectly correlated (i.e., they always yield the same value when measured in the computational basis); (2) the two bottom qubits are perfectly inversely correlated; (3) the two middle qubits are only partially correlated and perhaps only

partially entangled. However, this is not indicated at all in the single qubit reduced states which all indicate a purity of 0.5 in IBM's interface (or, equivalently, maximal mixedness in Quirk's interface). The *half-matrix*, presented in Section 4, is an alternative visual aid for analyzing circuits that makes it easier to understand such circuits.

3 VISUALIZING THE STATE VECTOR

3.1 Basic Display Options and Features

Figures 2, 5 and 6 show examples of state vectors whose amplitudes are shown using horizontal blue bars (whose horizontal length is related to the probability or magnitude, depending on the user's choice) and discs with a tick mark (to show phase). We now explain how the visual design of these amplitudes is based on principles from information visualization.

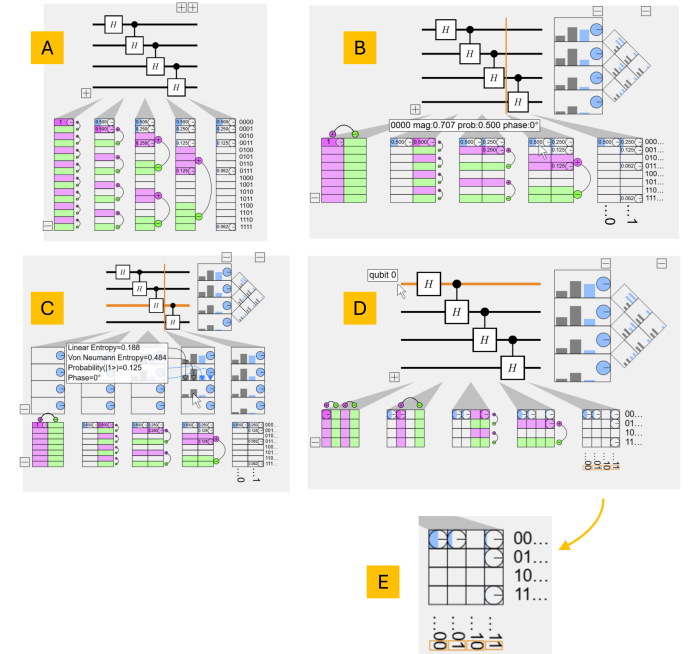


Fig. 5. The same circuit on 4 qubits is shown with different display options. The state vector for each layer has 16 amplitudes. A: each layer's state vector is shown as a 16×1 column vector. B: each state vector is wrapped into an 8×2 arrangement. C: the single-qubit reduced states are also shown for each layer. D: the amplitudes of the state vector are shown as squares, with blue bars behind each disc instead of to the left of each disc. E: enlarged view showing bits highlighted in orange corresponding to the top wire under the mouse cursor in D.

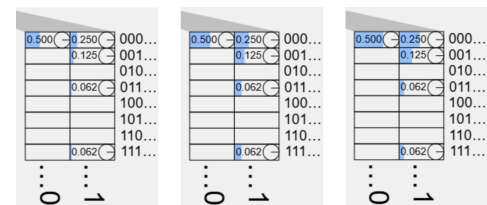


Fig. 6. The blue bars shown for each amplitude can have length proportional to probability (left), or proportional to magnitude of amplitude (middle), or be an appropriately scaled linear function of the logarithm of probability (right) to make differences between small values easier to see.

Probability or magnitude is shown using a bar length rather than, say, a shade of a color, because perceptual studies have established a hierarchy of visual channels (chapter 5 in [6]) that ranks length as one of the best ways to show a quantitative variable. The bars have text labels on top of them to indicate the numerical value of the probability. Thus, bar lengths allow a reader to quickly find large and small values, whereas the numerical text provides more precise information. The bars are oriented horizontally, rather than vertically, to make better use of the spatial resolution made available by the text. When an amplitude is zero, no disc is shown, to make it clearer that the amplitude is indeed zero and to make it more visually distinct from non-zero amplitudes. Phases are shown using the *orientation* of the disc's tick mark, rather than, say, the length of another bar, to reflect the fact that angles wrap around at 2π back to zero.

Each state vector can be wrapped into multiple columns to save space vertically (most of the state vectors in Figures 2, 5 and 6 are 8×2). Amplitudes can also be compressed into squares (Figure 5 D and E) to save even more space.

In many circuits, the magnitudes of the resulting amplitudes are very small and difficult to distinguish. We implemented different definitions of bar length (Figure 6) to allow increasing the visible difference between small values, making it easier to see which are bigger or smaller.

Hovering the mouse cursor over different parts of the user interface causes tooltips to appear (Figure 2 G, Figure 5 B and C) to make the user interface more self-revealing and provide more precise information. When appropriate, tooltips have curved callout arrows connecting them to corresponding visual elements (Figures 2 G, 5 C, 17, 18).

We also implemented multiple kinds of interactive coordinated highlighting [7]. Hovering over a state vector causes a vertical orange line segment to appear in the circuit between the corresponding layers (Figure 5 B). Hovering over reduced states causes the same vertical orange line segment to appear, and also highlights the corresponding wire in orange (Figure 5 C). Hovering over a wire causes the corresponding bits in the bitstrings to be highlighted (Figure 5 D and E; see also Figure 14). Finally, hovering over a cell in the half-matrix causes the corresponding *two* wires to be highlighted in orange (Figures 2 and 17).

3.2 State Vector Difference Highlighting

Consider a layer in a circuit where there is only one gate. The following sections consider various kinds of gates for which we can graphically show the effect on the state vector.

To explain the effect of the gate, we partition the amplitudes of the state vector into *even* and *odd* subsets. If there are n qubits, the amplitudes are identified by bitstrings of the form $b_{n-1} \dots b_0$. For example, if $n = 3$, the bitstrings are 000, 001, ..., 111. If the gate is on wire j , where $0 \leq j < n$, then the *even* amplitudes are defined as those whose bitstrings have their j th bit equal to 0, and similarly, the *odd* amplitudes have bitstrings with j th bit equal to 1. As an example, for $n = 3$, $j = 1$, the odd amplitudes have bitstrings 010, 011, 110, 111, where the j th bit is underlined. As another example, Figure 10 shows the even and odd amplitudes in different colors. We use these definitions of odd and even in the following sections.

3.2.1 Z gates

The effect of a Z on the state vector is to rotate the odd amplitudes. In our state vector difference highlighting, the subset of amplitudes affected by this is colored green.

Figure 7 shows the difference highlighting when the Z gate is on different wires. The green semicircular arrow to the right of each state vector indicates that the rotation angle is always π radians.

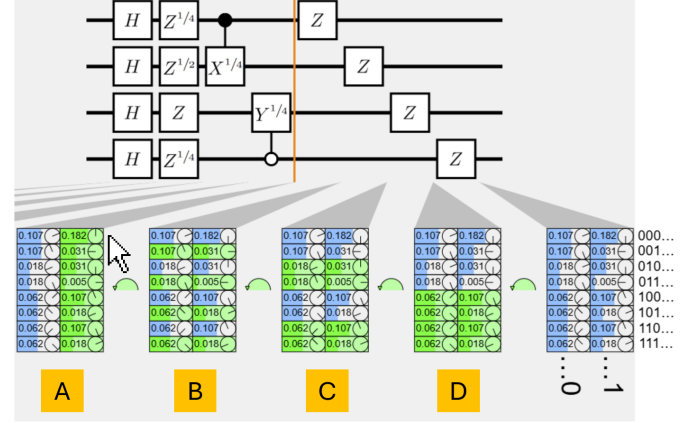


Fig. 7. Difference highlighting showing the effect of the Z gates in the last 4 layers of the circuit. Amplitudes rotated by each Z gate are colored green. (The first 4 layers of the circuit were designed to create a variety of amplitudes with different magnitudes and phases, to make it easier to see how these are modified in the subsequent layers. These first 4 layers are reused in several subsequent figures.)

If a Z gate has one or more control (or anticontrol) qubits, these simply limit the subset of amplitudes that are rotated. Examples of this are shown in Figure 8. Each additional control (or anticontrol) qubit reduces by half the subset of amplitudes that are colored green and that are rotated.

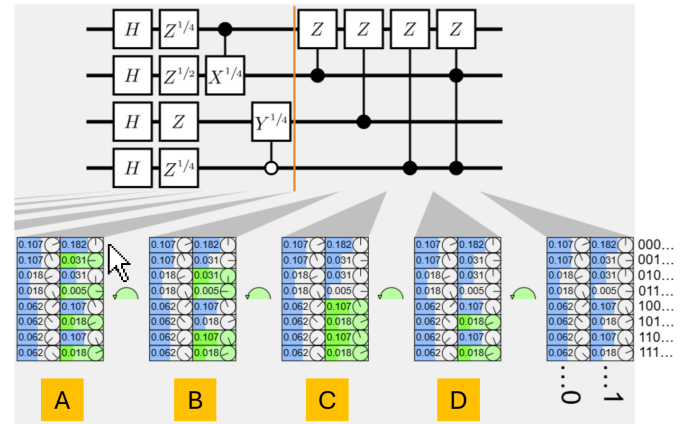


Fig. 8. The effect of controlled Z gates is restricted by control or anticontrol qubits. Notice that the subset of amplitudes colored green in D is the intersection of those in A and C.

3.2.2 S, T, Phase, and GlobalPhase gates

The difference highlighting for Z gates is easily extended to any single-qubit gates that rotate a subset of amplitudes, even with control or anticontrol qubits. We have done this in our software for $S = \sqrt{Z}$ and $T = \sqrt[4]{Z}$ and their inverses, as well as Z^k and $\text{Phase}(\theta) = Z^{\theta/\pi}$ (Figure 9). Our software

also supports a $\text{GlobalPhase}(\theta)$ gate, in which case the set of amplitudes colored green is the entire state vector, unless there are control (or anticontrol) qubits limiting the rotation to a subset.

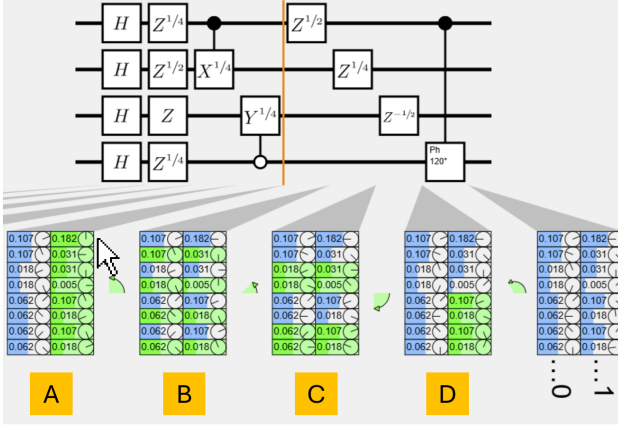


Fig. 9. Difference highlighting for other gates that rotate around the z axis of the Bloch sphere. A: $Z^{1/2} = S$. B: $Z^{1/4} = T$. C: $Z^{-1/2} = S^{-1}$. D: $\text{Phase}(120^\circ)$ with a control qubit.

3.2.3 X gates

The effect of an X gate is to exchange even and odd amplitudes of the state vector. To show this visually, we color the even and odd subsets in purple and green⁵, respectively (Figure 10). The double-headed curved arrows mean that each pair of contiguous purple and green blocks exchange places.

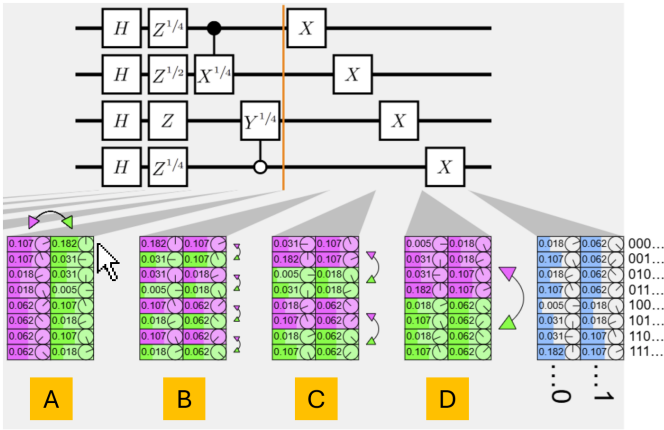


Fig. 10. The effect of X gates on different wires. Even amplitudes are highlighted in purple, odd amplitudes in green.

Similar to the previous examples, adding control qubits simply restricts the effect of the X gate to a smaller subset of amplitudes (Figure 11).

3.2.4 Y gates

Visualizing the effect of Y gates is more difficult. However, we can notice that $Y = SX S^{-1} = Z^{1/2} X Z^{-1/2}$. Reading

5. These colors were chosen because most forms of colorblindness involve a difficulty or inability to distinguish along the red-green dimension or the yellow-blue dimension of color space, but purple and green differ along *both* of these dimensions simultaneously.

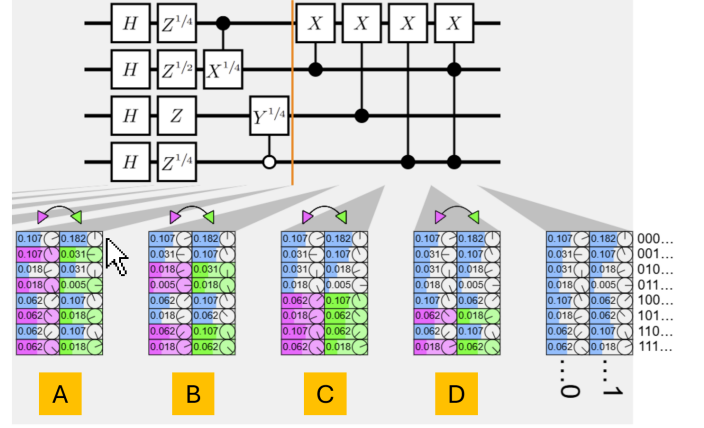


Fig. 11. Controlled X gates. Notice that the subset colored purple and green in layer D is the intersection of the subsets in layers A and C.

this right-to-left, $Z^{-1/2}$ rotates the odd amplitudes by $-\pi/2$ radians, then X exchanges the even and odd amplitudes, and finally, $Z^{1/2}$ rotates the amplitudes that were originally even by $+\pi/2$ radians. This is equivalent to rotating the even and odd subsets by $+\pi/2$ and $-\pi/2$, respectively, followed by an exchange. Thus, we can combine and extend the visual feedback already designed for the $S^{\pm 1} = Z^{\pm 1/2}$ and X gates, yielding two rotation angles (one for each colored subset) and double-headed curved “exchange” arrows (Figure 12).

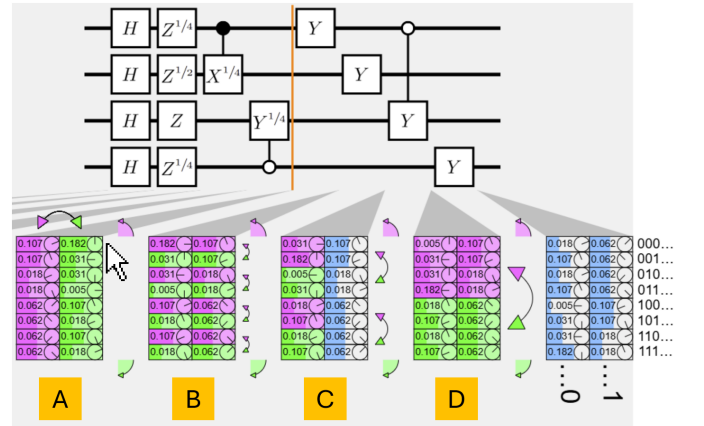


Fig. 12. Difference highlighting for Y gates. For each Y gate, imagine the $+\pi/2$ purple anticlockwise rotation applied to the purple (even) subset, the $-\pi/2$ green clockwise rotation applied to the green (odd) subset, and then the purple and green subsets exchanging places to yield the state vector in the next layer.

3.2.5 Hadamard gates

For Hadamard gates, the concept of even and odd subsets applies again, but now these are being added and subtracted and scaled by $1/\sqrt{2}$. The addition and subtraction is represented by the $\oplus$ and $\ominus$ symbols (Figure 13), and the scaling factor is implicit. The intention is to show how multiple Hadamard gates can distribute a single non-zero amplitude over many base states (Figure 14, first three layers of each of the circuits) and also used to cancel amplitudes (Figure 14, last three layers of each of the circuits).

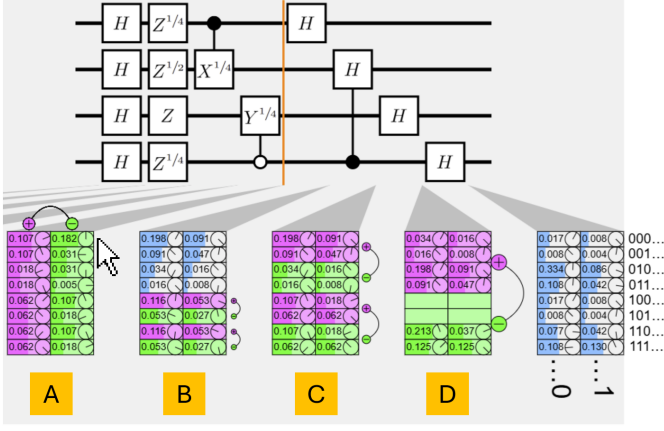


Fig. 13. Difference highlighting for Hadamard gates. For each H gate, imagine the purple and green subsets being added ($\oplus$) and subtracted ($\ominus$), with the results replacing the purple and green subsets in the next layer, respectively, with an implicit scaling factor.

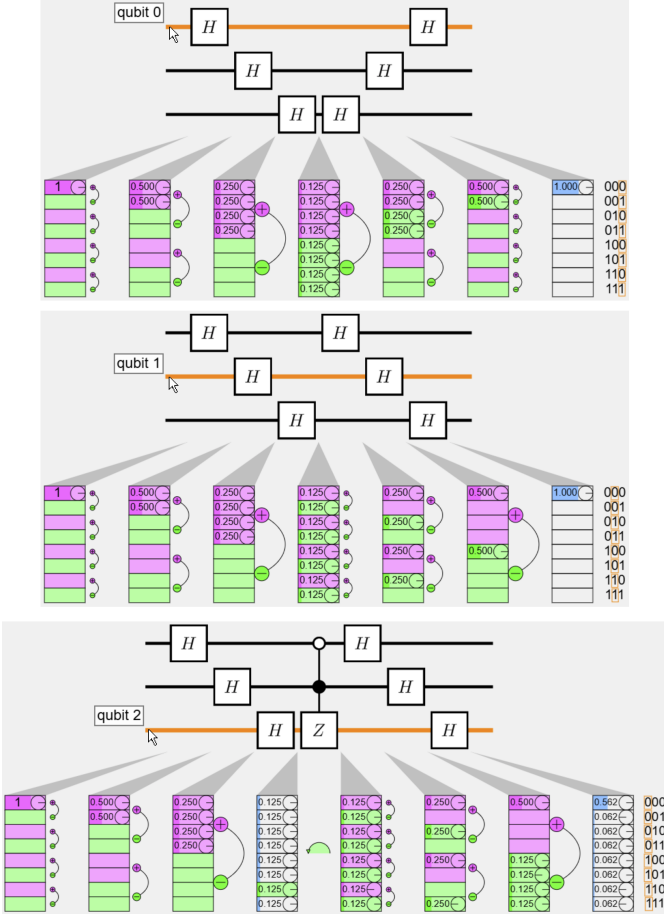


Fig. 14. In each circuit, in the first three layers, the H gates split and distribute the nonzero amplitudes from purple to green amplitudes. In the last three layers, the subtractions cancel out the green amplitudes. Each screenshot shows a different wire and bitstring bit highlighted in orange, as a reminder of their association. Also, as with all screenshots throughout the article, the numeric labels on the state vectors show probabilities, not magnitudes of amplitudes.

To better understand Figure 14, keep in mind that the numeric labels on the state vectors show probabilities, and let p_e and p_o be the probabilities for two matching base

states, where the first is even (purple) and the second is odd (green). If we assume that the phases of the relevant amplitudes are oriented in the same direction, then the probability values of (p_e, p_o) , under the effect of a Hadamard gate, will be replaced in the next layer with $(\frac{1}{2}(\sqrt{p_e} + \sqrt{p_o})^2, \frac{1}{2}(\sqrt{p_e} - \sqrt{p_o})^2)$. Now consider three cases.

Case 1: $p_o = 0$, as in the first 3 layers of the circuits in Figure 14. In this case, each $(p_e, 0)$ pair is replaced with $(\frac{1}{2}(\sqrt{p_e} + \sqrt{0})^2, \frac{1}{2}(\sqrt{p_e} - \sqrt{0})^2) = (\frac{1}{2}p_e, \frac{1}{2}p_e)$, spreading the probability evenly between the two base states.

Case 2: we have the same probability $p_e = p_o$ on both base states, and the relevant phases have the same orientation, as seen in the last 3 layers of the top and middle circuits of Figure 14. Doing the calculation, we find that each (p_e, p_e) pair is replaced with $(2p_e, 0)$, concentrating the probability on the even base state.

Case 3: $p_e = p_o$, but the relevant phases are π radians different, as seen in one case in the bottom circuit of Figure 14, immediately after the Z gate, for the base states 110 and 111. In this case, we calculate that (p_e, p_e) is replaced with $(0, 2p_e)$, concentrating probability on the odd base state.

3.2.6 SWAP gates

To understand the effect of a SWAP gate on the state vector, consider as an example that there are $n = 4$ qubits, hence $2^n = 16$ base states represented by all the binary strings of length 4, and we can write these strings as $b_3b_2b_1b_0$ where b_k is the k th binary bit. Assume also that there is a SWAP gate acting on qubits i and j , where $0 \leq i < j < n$. To take a concrete example, assume $i = 1$ and $j = 3$. Then, the amplitudes for the following pairs of base states will be swapped: $(\underline{00}10, \underline{10}00)$, $(\underline{01}11, \underline{11}01)$, and any pair of the form $(\underline{0b_21b_0}, \underline{1b_20b_0})$, where we have underlined bits b_i and b_j .

In our visualization, we use double-headed arrows to show which amplitudes are swapped. Assume that the state vectors are wrapped and displayed as 2^{n-K} rows $\times$ 2^K columns (in Figure 15, $n = 4$ and $K = 1$). In each layer, there are three possible cases. If $K \leq i < j$, then each pair of amplitudes that are swapped appear in the same column (Figure 15, layers C, D). If $i < K \leq j$, then each pair of amplitudes that are swapped appear in diagonally located positions (Figure 15, layers A, B). If $i < j < K$, then each pair of amplitudes that are swapped appear in the same row (not shown).

3.2.7 Visual universality

We have demonstrated how a small set of visual primitives, namely the coloring of even and odd subsets (in purple and green), curved arrows for rotation angles, and symbols for exchanging (double-headed arrows), adding and subtracting ($\oplus$ and $\ominus$), allow us to elucidate the effect of the following core set of gates: Hadamard, X , Y , Z , $S = \sqrt{Z}$, S^{-1} , $T = \sqrt[3]{Z}$, T^{-1} , Z^k , $\text{Phase}(\theta)$, $\text{GlobalPhase}(\theta)$, and SWAP, each with arbitrary control and anticontrol qubits. This includes, by the way, Toffoli and Fredkin gates. However, ideally we would support arbitrary gates. To articulate this goal, we propose a definition: a set of visual primitives is *visually universal*, with respect to a set of objects, if any of those objects can be visualized using the primitives. Thus, we seek visual universality for our set of primitives, with

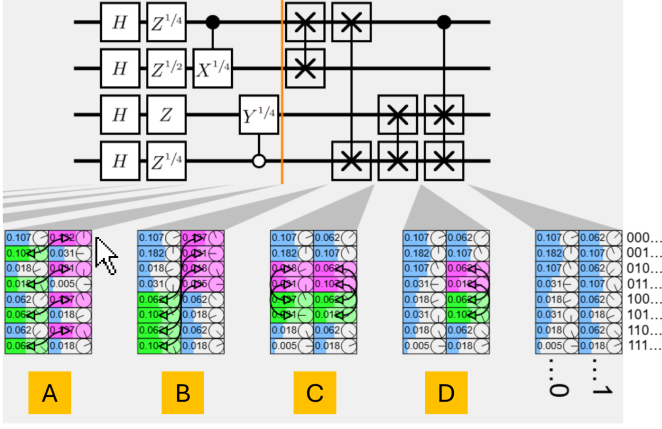


Fig. 15. Difference highlighting with SWAP and controlled-SWAP gates.

respect to the set of single-qubit unitary gates with arbitrary control and anticontrol qubits.

It turns out that any gate not in our core set can be rewritten as a product of gates from the core set. By replacing individual gates outside the core set with sequences of core gates, we can visualize the effects on the state vector, at the cost of increasing the depth of the circuit. Below we list various gates and their equivalent expansions, with their *cost*, defined as the number of layers required to replace each original gate. Note that, in many applications, a designer might not be interested in changes to global phase, in which case GlobalPhase gates can be removed from the expansions, saving one layer.

Gate	Expansion into core gates	Cost
$X^{1/2}$	HS	3
$X^{-1/2}$	$HS^{-1}H$	3
$Y^{1/2}$	$HZ\text{GlobalPhase}(\pi/4)$	3
$Y^{-1/2}$	$ZH\text{GlobalPhase}(-\pi/4)$	3
X^k	HZ^kH	3
Y^k	$HS^{-1}HZ^kHS$	7
$RX(\theta)$	$HZ^{\theta/\pi}H\text{GlobalPhase}(-\theta/2)$	4
$RY(\theta)$	$HS^{-1}HZ^{\theta/\pi}HS\text{GlobalPhase}(-\theta/2)$	8
$RZ(\theta)$	$Z^{\theta/\pi}\text{GlobalPhase}(-\theta/2)$	2

It is also well known that *any* single-qubit unitary gate can be written as a product of appropriate rotation gates, and there exist algorithms to convert any multi-qubit gate into a sequence of multi-controlled single-qubit gates [8], [9], establishing the visual universality that we sought.

3.2.8 Generalized gates enabling shorter expansions

The visual feedback shown for Y gates (§ 3.2.4) is the most complicated presented so far, since it combines *two* rotation angles with a double-headed arrow, and yet the rotation angles are fixed at $\pm\pi/2$ radians. This implies that we are not making full use of the degrees of freedom available in our set of visual primitives. We therefore propose three new gates that allow these rotation angles to be independently varied and combined with the double-headed arrow and the addition and subtraction symbols. Although these new gates are not standard, their effect on the state vector is easy to understand using our set of primitives, and they also

allow gates outside the core set to be expanded into *shorter* equivalent sequences of core gates.

We now define the three new gates to be added to the core set, which we call *Generalized Z*, *Generalized Y*, and *Generalized Hadamard* gates. Each takes two angle parameters a, b :

$$Z_G(a, b) = X \cdot Z^{a/\pi} \cdot X \cdot Z^{b/\pi} = \begin{bmatrix} e^{ia} & 0 \\ 0 & e^{ib} \end{bmatrix}$$

$$Y_G(a, b) = Z^{a/\pi} \cdot X \cdot Z^{b/\pi} = \begin{bmatrix} 0 & e^{ib} \\ e^{ia} & 0 \end{bmatrix}$$

$$H_G(a, b) = H \cdot Z_G(a, b) = \frac{1}{\sqrt{2}} \begin{bmatrix} e^{ia} & e^{ib} \\ e^{ia} & -e^{ib} \end{bmatrix}$$

To gain intuition for how these gates work, read the definition of the Z_G gate from right-to-left: first, the $Z^{b/\pi}$ rotates the odd amplitudes by angle b ; next, the X exchanges the even and odd amplitudes; next, the $Z^{a/\pi}$ rotates the amplitudes that were originally even by angle a ; and finally, the left-most X exchanges the amplitudes again. The net effect is that the even and odd amplitudes are rotated by angles a and b , respectively. The Y_G gate works similarly, except it leaves the even and odd amplitudes exchanged. And the H_G gate applies a Hadamard after rotating the even and odd subsets.

These generalized gates are visualized as shown in Figure 16, and they enable the following improved, shorter expansions of non-core gates:

Gate	Expansion into core gates using generalized gates	Cost
$X^{1/2}$	$H_G(0, \pi/2)H$	2
$X^{-1/2}$	$H_G(0, -\pi/2)H$	2
$Y^{1/2}$	$H_G(\pi/4, 5\pi/4)$	1
$Y^{-1/2}$	$Z_G(-\pi/4, -5\pi/4)H$	2
X^k	$H_G(0, k\pi)H$	2
Y^k	$H_G(0, -\pi/2)H_G(0, k\pi)H_G(0, \pi/2)H$	4
$RX(\theta)$	$H_G(0, \theta)H_G(-\theta/2, -\theta/2)$	2
$RY(\theta)$	$H_G(0, -\pi/2)H_G(0, \theta)H_G(0, \pi/2)H_G(-\theta/2, -\theta/2)$	4
$RZ(\theta)$	$Z_G(-\theta/2, \theta/2)$	1

Users of MuqcsCraft can press a special “Expand Circuit” button to invoke automatic replacement of the non-core gates in the tables above with core gates, using generalized gates or not in the resulting expansion. This expansion makes the circuit deeper but also allows difference highlighting to show the effects of all gates on the state vector.

4 QUBIT PAIRS: THE HALF-MATRIX

A triangular half-matrix is one way to organize pairs of elements, and has been used in previous visualizations [10], [11] not related to quantum computing. Each cell of our half-matrix corresponds to a pair of qubits, as revealed when the mouse hovers over a cell and the corresponding qubit wires highlight in orange (Figure 17).

Figure 17 shows our half-matrix to the right of the circuit diagram. The circuit shown is very similar to that in Figure 1. In both figures, the output state has high probabilities for base states 0111 and 1000, low probabilities for 0100 or 1011, and zero for all others. Our half-matrix in Figure 17 reveals that the correlation (in the computational basis) of the top two qubits is $+1$, that of the bottom two qubits is

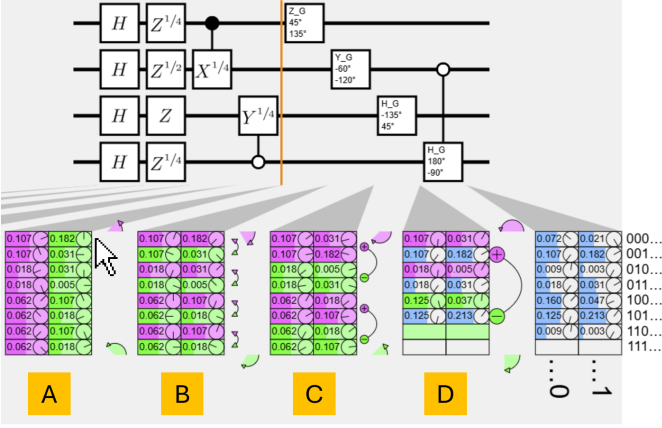


Fig. 16. Difference highlighting for the generalized Z_G , Y_G , H_G gates. For each of these gates, imagine the purple rotation applied to the purple subset, and the green rotation applied to the green subset. Then, in the case of the Y_G gate (layer B), the subsets exchange places. In the case of the H_G gate (layers C and D), imagine the purple and green subsets being added ($\oplus$) and subtracted ($\ominus$), with the results replacing the purple and green subsets, respectively, with an implicit scaling factor.

−1, and that of the middle two qubits is 0.707. We might be tempted to conclude that there is a stronger entanglement between the top two and bottom two qubits, and weaker entanglement between the middle two. However, correlation is not a proper metric of entanglement, since it depends on the choice of basis. The half-matrix also shows concurrence [4] values, and reveals the top two and bottom two qubits to have a concurrence of 0.707, while the middle two have a concurrence of zero. This is not shown at all in the user interfaces of Figure 1.

In Figure 17, top and middle, each cell of the half-matrix shows bars revealing the metrics we chose to display. Metrics related to mixedness are shown in dark grey, and those related to the intra-pair relationship are shown in blue if positive or red if negative. Additional metrics could, of course, be added as desired. We also experimented with showing the metrics as rectangular glyphs [12], shown in Figure 17, bottom. The two metrics related to mixedness are mapped to the orthogonal dimensions of a dark grey rectangle, and the two intra-pair metrics are mapped to orthogonal dimensions of another rectangle whose sides are blue or red to indicate the sign of the metric. A user accustomed to this visual feedback could plausibly quickly find rectangles of interest, based on their size and color, in a large half-matrix.

5 ADDITIONAL EXAMPLES

Figure 18 shows a circuit that generates a W-4 state, where the four base states with a single 1 in their bitstring (namely 0001, 0010, 0100, and 1000) are the only ones with non-zero probability. For each layer, we see (Figure 18 A) information about the single-qubit reduced states, and (Figure 18 B) state vector difference highlighting. Users who learn how to read (B) can use it to design changes to the circuit, or to correct small errors in the placement of gates, since each gate is directly related to the repositioning or rotation of amplitudes. However, (A) provides much less guidance and establishes a much less direct relationship between gates

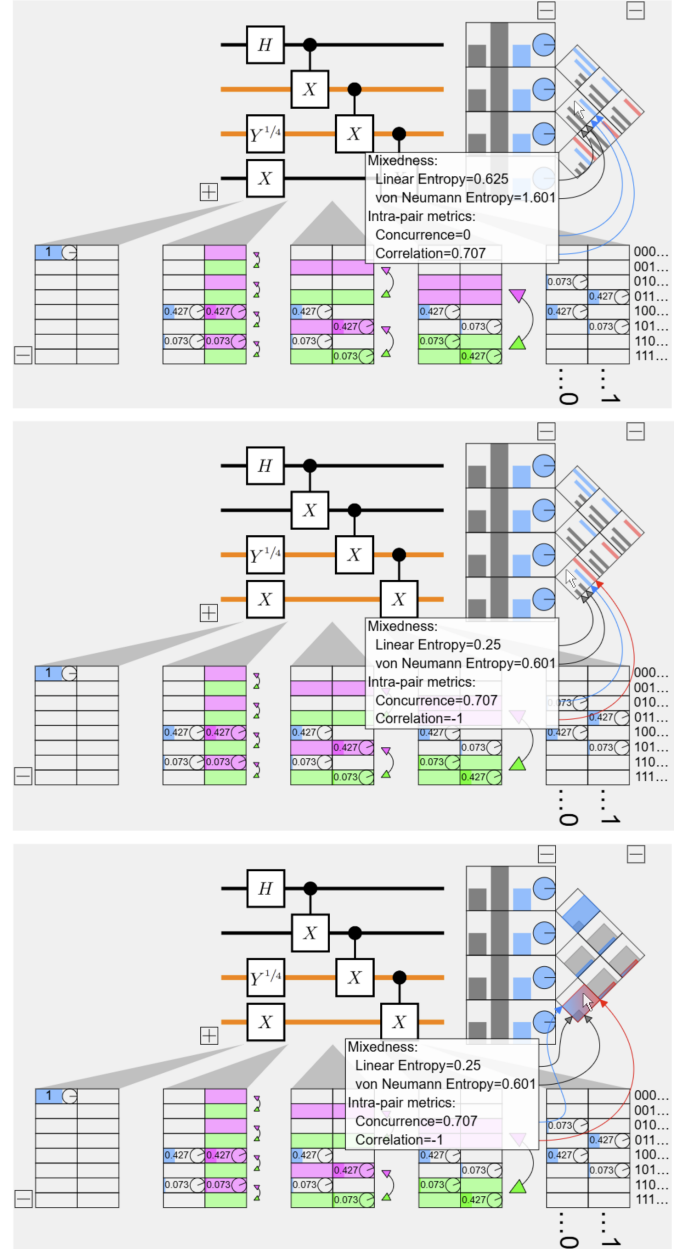


Fig. 17. Top: the mouse cursor hovers over the middle-left cell of the half-matrix, which corresponds to the two middle qubits, whose wires are highlighted in orange. The tooltip reveals that these two qubits have correlation (in the computational basis) of 0.707, and concurrence 0. Middle: the bottom two qubits have correlation −1 and concurrence 0.707. Bottom: the cells of the half-matrix display rectangular glyphs instead of bars, which might make it easier to quickly find cells with high or low levels of mixedness or correlation / concurrence.

and the evolution of the state. In fact, part of the inspiration for designing the visual feedback in (B) came from one of us trying to design a circuit in Quirk, through trial-and-error, to generate a W-4 state, and realizing that gates like a multi-control X gate (also called generalized Toffoli gate) can move non-zero amplitudes around the state vector, and that new visual feedback could be used to guide the user in the choice and placement of such gates.

Figure 19 shows essential parts of Grover’s algorithm: how the H gates in the initialization spread probability evenly over all base states, how the oracle rotates a single

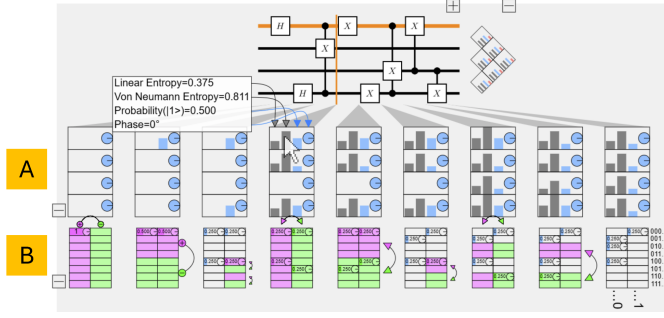


Fig. 18. A circuit generating a W-4 state. Below the circuit diagram, for each layer, we see (A) the reduced state for each qubit, and (B) the state vector with difference highlighting. Which of those two seems more useful for understanding the operation of the circuit?

amplitude, and how the subsequent H gates concentrate probability onto the marked base state.

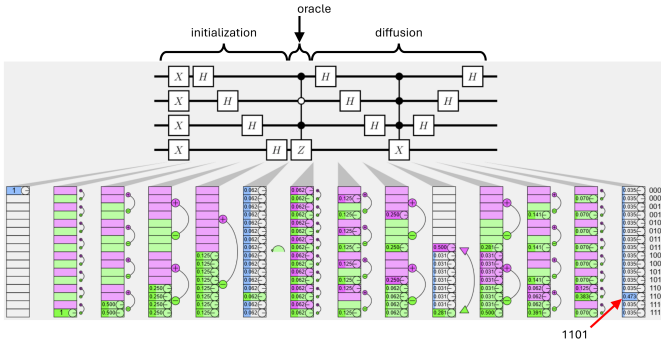


Fig. 19. A circuit implementing a single iteration of Grover's algorithm, increasing the probability of the amplitude for one base state. The oracle marks one base state by rotating its amplitude.

6 IMPLEMENTATION DETAILS

MuqcsCraft [5] is built on top of the open-source Muqcs library [13]. The circuit is simulated using a state vector simulator, where the state vector is updated layer-by-layer using an efficient implementation of qubit-wise multiplication (§3 in [1]). To compute quantities related to reduced states, an efficient partial trace subroutine (§6 in [1]) computes all single-qubit 2×2 reduced density matrices, from which qubit probability, phase, purity, and other quantities are computed. To populate the half-matrix, we compute all two-qubit 4×4 reduced density matrices, from which pairwise concurrences and other quantities are computed.

Like Quirk [3], the gates in the circuit are encoded in the URL query string of the MuqcsCraft web application, making it possible to bookmark circuits in a web browser and share hyperlinks to circuits in plain text documents. Because of this, the web browser's Back button provides a way to undo changes to the circuit. There are also features to easily export circuits from MuqcsCraft to Quirk and to IBM Quantum Composer [2].

7 LIMITATIONS

Our state vector difference highlighting requires that each gate appear in its own layer, is limited to ≈ 8 qubits,

and only supports a set of “core” gates. Although any unitary gate can be replaced with an equivalent sequence of core gates whose effects can be visualized, in practice these sequences are sometimes long, which could impede understanding of their cumulative effect.

8 RELATED WORK

8.1 Visualizations for Quantum Computing

A growing body of work applies principles of information visualization [6] and interaction design to quantum computing, e.g., showing higher-level blocks in a circuit [14], allowing users to sketch circuits [15], visualizing and manipulating tensor networks [16], visualizing noise [17] and graph states [18], or enabling a better understanding of Grover's algorithm [19], Shor's algorithm [20], quantum machine learning [21], and variational quantum algorithms (VQAs) [22].

We now review topics related more specifically to our current work.

8.2 Visualization of circuits

Compared to previous work, MuqcsCraft is the only graphical circuit simulator that highlights differences in the state vector from one layer to the next, as well as the only to display a triangular half-matrix, and the only one that computes and shows pairwise concurrence values to aid in understanding entanglement. Below we discuss previous work and some of their specific features.

8.2.1 Popular graphical interfaces

IBM Quantum Composer [2] and Quirk [3] are both web-based, like MuqcsCraft. Quantum Composer is limited to 4 qubits without an account and is closed source, whereas Quirk and MuqcsCraft support 16 qubits and are both open source. Quantum Composer lacks support for anticontrol qubits, lacks support for control qubits on certain gates (T and S), supports a maximum of one control qubit on most gates, and requires several clicks to add this control qubit, whereas Quirk and MuqcsCraft allow arbitrary control and anticontrol qubits to be added to any gate with a simple drag-and-drop. Quantum Composer also has no support for continuously varying the parameters of gates, whereas Quirk supports animated rotation gates, and MuqcsCraft allows parameters of gates to be smoothly varied by dragging with the mouse which triggers continuously updated visual feedback.

To visualize the evolution of the single-qubit reduced states, both Quantum Composer and Quirk allow widgets to be drag-and-dropped onto the circuit between each layer of gates, however MuqcsCraft allows the visibility of reduced states for all layers to be toggled with a single click.

To visualize the evolution of the state vector, Quantum Composer allows the user to step through the circuit, one layer at a time, while a single colored barchart of the state vector is updated. Quirk and MuqcsCraft both display the state vector with wrapping to save space (and in MuqcsCraft, wrapping is an option that can be toggled). Quirk allows state vector widgets to be drag-and-dropped onto the circuit between each layer of gates. MuqcsCraft instead

allows the visibility of state vectors for all layers to be toggled with a single click, and is the only package to highlight the differences from one layer to the next.

Comparing the two open-source packages, Quirk is over 40k lines of code versus MuqcsCraft’s less than 10k making it easier to study and understand. Quirk uses GPU acceleration to simulate circuits faster, while MuqcsCraft avoids the complexity of GPU programming and nevertheless simulates circuits of 16 qubits or less in under 10 ms per gate on a 2022 laptop.

8.2.2 Academic work on circuit visualization

Rainbow boxes [23] represent qubits with rectangles, where height shows probability, color shows phase, and entanglement is indicated by merging rectangles, however this can only be done for adjacent qubits, and makes no distinction between qubits that are partially versus maximally entangled.

QuFlow [24] shows causal relationships between amplitudes from one layer to the next, and QuantumEyes [25] has a Probability Summary View that shows the evolution of probabilities of base states layer-by-layer. However, neither of these use wrapping to make the state vectors consume less vertical space, making them less scalable than Quirk and MuqcsCraft. In addition, there is no obvious way to modify these techniques to support wrapping.

Also, unlike these previous works, we provide the source code for MuqcsCraft and a working interactive online demo.

8.3 Extensions to the Bloch sphere

Methods for embedding sets of quantum states in space can be developed through the perspective of abstract algebra and topology, discussed in depth by Bengtsson and Życzkowski [26]. Previous works have extended the Bloch sphere to two qubits [27], [28], [29], e.g., to better understand entanglement, and even to multiple qubits [30], [31], [32], however these extensions require 3D geometry that is difficult to visualize, and/or have limited scalability to many qubits.

In contrast, the triangular half-matrix used in MuqcsCraft is purely 2D, and occupies an area that scales quadratically with the number of qubits, allocating one cell for each pair of qubits to show information about how the pair are related.

9 CONCLUSIONS AND FUTURE DIRECTIONS

We presented two novel visualization techniques for quantum circuits, state vector difference highlighting and the triangular half-matrix, demonstrated in our open-source MuqcsCraft software. Difference highlighting uses a small set of visual primitives to illustrate the effect of each gate, and is visually universal over single-qubit gates with control and anticontrol qubits, if we substitute certain gates with equivalent sequences. The half-matrix provides pairwise information about qubits, revealing entanglement and correlation patterns that are not visible in common single-qubit reduced state displays. This work is a further step toward tools for understanding and designing quantum algorithms.

In future work, difference highlighting might be modified to be applicable to subsets of qubits, making our system

more scalable to large circuits; or extended to illustrate the aggregated effect of multi-qubit gates or higher-level blocks of gates. For example, visualizing the aggregated effect of the diffuser in Grover’s algorithm, iterated a few times, would help to illustrate the algorithm. It could also be fruitful to consider changes of basis in how the state vector is visualized, making certain gates easier to understand. The half-matrix might be modified to allow wires to be reordered [33] to cluster interesting subsets of qubits, or only show the most interesting pairs of qubits and occupy less space.

ACKNOWLEDGMENTS

Thanks to Thomas R. Bromley who gave advice and implemented a version of the pairwise concurrence computation for the half-matrix, and to Olivier Landon-Cardinal who gave suggestions for how to evaluate the software prototype. This research was supported financially by NSERC and by Alicia Servera.

REFERENCES

- [1] M. J. McGuffin, J.-M. Robert, and K. Ikeda, “How to write a simulator for quantum circuits from scratch: A tutorial,” 2025, <https://arxiv.org/abs/2506.08142>.
- [2] IBM, “IBM quantum composer,” 2023, <https://quantum.cloud.ibm.com/composer>.
- [3] C. Gidney, “Quirk,” 2020, <https://algassert.com/quirk>.
- [4] V. Coffman, J. Kundu, and W. K. Wootters, “Distributed entanglement,” *Physical Review A*, vol. 61, no. 5, p. 052306, 2000.
- [5] M. J. McGuffin, “MuqcsCraft: A web-based graphical simulator and visualizer for quantum circuits,” 2025, <https://github.com/MJMcGuffin/MuqcsCraft>.
- [6] T. Munzner, *Visualization Analysis and Design*. CRC press, 2014.
- [7] S. Tan, C. Lai, X. L. Zhang, and X. Yuan, “An empirical guide for visualization consistency in multiple coordinated views,” in *IEEE Pacific Visualization Symposium (PacificVis)*, 2023, pp. 31–40.
- [8] M. Möttönen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, “Quantum circuits for general multiqubit gates,” *Physical review letters*, vol. 93, no. 13, p. 130502, 2004.
- [9] V. Shende, S. Bullock, and I. Markov, “Synthesis of quantum-logic circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 6, pp. 1000–1010, 2006.
- [10] L. Wilkinson, A. Anand, and R. Grossman, “Graph-theoretic scagnostics,” in *IEEE Symp. Information Visualization (InfoVis)*, 2005.
- [11] J.-F. Im, M. J. McGuffin, and R. Leung, “GPLOM: the generalized plot matrix for visualizing multidimensional multivariate data,” *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2013.
- [12] R. Borgo, J. Kehrler, D. H. Chung, E. Maguire, R. S. Laramée, H. Hauser, M. Ward, and M. Chen, “Glyph-based visualization: Foundations, design guidelines, techniques and applications,” in *Eurographics STARs (State of the Art Reports)*, 2013, pp. 39–63.
- [13] M. J. McGuffin, “Muqcs.js: McGuffin’s useless quantum circuit simulator,” 2025, <https://github.com/MJMcGuffin/muqcs.js>.
- [14] Z. Wen, Y. Liu, S. Tan, J. Chen, M. Zhu, D. Han, J. Yin, M. Xu, and W. Chen, “Quantivine: A visualization approach for large-scale quantum circuit representation and analysis,” *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2023.
- [15] I. Arawjo, A. DeArmas, M. Roberts, S. Basu, and T. Parikh, “Notational programming for notebook environments: A case study with quantum circuits,” in *ACM Symp. User Interface Software and Technology (UIST)*, 2022, pp. 1–20.
- [16] A. Kissinger and V. Zamdzhiev, “Quantomatic: A proof assistant for diagrammatic reasoning,” in *CADE-25: 25th International Conference on Automated Deduction*. Springer, 2015, pp. 326–336.
- [17] S. Ruan, Y. Wang, W. Jiang, Y. Mao, and Q. Guan, “VACSSEN: A visualization approach for noise awareness in quantum computing,” *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 29, no. 1, pp. 462–472, 2023.

- [18] M. Miller and D. Miller, "GraphStateVis: Interactive visual analysis of qubit graph states and their stabilizer groups," in *IEEE Int. Conf. Quantum Computing and Engineering (QCE)*, 2021, pp. 378–384, <https://github.com/graphstatevis/app>.
- [19] S. Norrie, A. Estey, H. Müller, and U. Stege, "QGrover: Teaching grover's algorithm through visual exploration," in *IEEE Int. Conf. Quantum Computing and Engineering (QCE)*, vol. 3, 2024, pp. 17–24.
- [20] Z. Tao, Y. Pan, A. Chen, and L. Wang, "ShorVis: A comprehensive case study of quantum computing visualization," in *Int. Conf. Virtual Reality and Visualization (ICVRV)*. IEEE, 2017, pp. 360–365.
- [21] S. Ruan, Z. Liang, Q. Guan, P. Griffin, X. Wen, Y. Lin, and Y. Wang, "VIOLET: Visual analytics for explainable quantum neural networks," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 30, no. 6, pp. 2862–2874, 2024.
- [22] M. S. Rudolph, S. Sim, A. Raza, M. Stechly, J. R. McClean, E. R. Anschuetz, L. Serrano, and A. Perdomo-Ortiz, "ORQVIZ: Visualizing high-dimensional landscapes in variational quantum algorithms," 2021, <https://arxiv.org/abs/2111.04695>.
- [23] J.-B. Lamy, "Dynamic software visualization of quantum algorithms with rainbow boxes," in *International Conference on Information Visualization Theory and Applications (IVAPP)*, 2019.
- [24] S. Lin, J. Hao, and L. Sun, "QuFlow: Visualizing parameter flow in quantum circuits for understanding quantum computation," in *IEEE Scientific Visualization Conference (SciVis)*, 2018, pp. 37–41.
- [25] S. Ruan, Q. Guan, P. Griffin, Y. Mao, and Y. Wang, "QuantumEyes: Towards better interpretability of quantum circuits," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2023, <https://quantumeyes.github.io/>.
- [26] I. Bengtsson and K. Życzkowski, *Geometry of quantum states: an introduction to quantum entanglement*, 2nd ed. Cambridge University Press, 2017.
- [27] O. Gamel, "Entangled bloch spheres: Bloch matrix and two-qubit state space," *Physical Review A*, vol. 93, no. 6, p. 062320, 2016.
- [28] C.-R. Wie, "Two-qubit bloch sphere," *Physics*, vol. 2, no. 3, pp. 383–396, 2020.
- [29] L.-H. H. Chang, S. Roccaforte, Z. Xu, and P. Cadden-Zimansky, "Geometric visualizations of single and entangled qubits," *American Journal of Physics*, vol. 92, no. 7, pp. 528–537, 2024.
- [30] J. B. Altepeter, E. R. Jeffrey, M. Medic, and P. Kumar, "Multiple-qubit quantum state visualization," in *Conf. Lasers and Electro-Optics and Int. Quantum Electronics Conf. (CLEO/IQEC)*. IEEE, 2009, pp. 1–2.
- [31] H. Mäkelä and A. Messina, "N-qubit states as points on the bloch sphere," *Physica Scripta*, vol. 2010, no. T140, p. 014054, 2010.
- [32] J. Bley, E. Rexigel, A. Arias, N. Longen, L. Krupp, M. Kiefer-Emmanouilidis, P. Lukowicz, A. Donhauser, S. Küchemann, J. Kuhn, and A. Widera, "Visualizing entanglement in multiqubit systems," *Physical Review Research*, vol. 6, no. 2, p. 023077, 2024.
- [33] M. Behrisch, B. Bach, N. H. Riche, T. Schreck, and J.-D. Fekete, "Matrix reordering methods for table and network visualization," *Computer Graphics Forum*, 2016, <http://matrixreordering.dbvis.de/>.



Jean-Marc Robert is a full professor at ETS where his students do research in IT security and quantum cryptography. His recent interests include quantum computing and quantum information theory. In 1985, he completed his Master's degree, formalizing the concept of privacy amplification, which led to the second paper ever published on quantum cryptography.



Michael J. McGuffin is a full professor at ETS, a French-language engineering school in Montreal, Canada, where his students do research in visualization and HCI. In 2009, his paper at the IEEE Information Visualization Conference (InfoVis 2009) received an Honorable Mention.