K. Feichtinger, L. Sonnleithner, H. Hajiabadi (Hrsg.): SE 2025 – Companion Proceedings,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2025

Improved Bounds for Rounding Errors in Quantum Circuit Simulators [4]

Jonas Klamroth[1], Niko Lemke[1], Ruben Götz[2], and Bernhard Beckert[3]

Abstract: Simulators play a crucial role in the development of quantum software, yet they differ from actual quantum devices in that their computations are carried out using floating-point arithmetic rather than real arithmetic. In [7], we introduced bounds on the errors that may arise due to these discrepancies. In the present work, we extend and refine these bounds, demonstrating that we can achieve tighter estimates that scale more efficiently with the number of gates in a quantum circuit. Furthermore, the newly derived bounds can be computed with minimal relaxation, making them practically applicable. We show that these improved bounds are effective in excluding significant errors across a wide range of quantum circuits, thus providing a more reliable framework for simulating quantum systems.

Keywords: quantum circuit simulation, floating-point errors, numeric analysis

1 Introduction

Motivation Quantum computers hold undeniable potential, theoretically offering a superpolynomial speedup compared to classical solutions. However, current quantum devices are far from achieving this theoretical speedup in practice due to their limited qubit counts and susceptibility to NISQ (Noisy Intermediate-Scale Quantum) errors. Consequently, much of the development in quantum computing is – and likely will be for the foreseeable future – conducted on simulators rather than real quantum devices. This is not only due to unreliable hardware but also because quantum simulators offer features like intermediate state examination and debugging, which are physically impossible on real devices.

Nevertheless, the use of simulators comes with a caveat. Quantum simulators are implemented in classical languages and thus rely on finite precision floating-point arithmetic to simulate quantum computations. This stands in stark contrast to real quantum devices, which operate in a real-valued domain. It is well established that floating-point arithmetic introduces rounding errors that can lead to inaccurate results. However, in many cases, these errors can be characterized and bounded, providing assurance that the error in a calculation will not exceed a certain threshold. This work tackles the question of whether the

[1] FZI Research Center for Information Technology, Software Engineering, Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germnay, klamroth@fzi.de; lemke@fzi.de

[2] Karlsruhe Institute for Technology, Kaiserstr. 12, 76131 Karlsruhe, Germany, uzgrr@student.kit.edu

[3] Karlsruhe Institute for Technology, Application-Oriented Formal Verification, Kaiserstr. 12, 76131 Karlsruhe, Germany, beckert@kit.edu

[4] This work is part of the KQCBW24 project funded by the Ministry of Economic Affairs Baden-Württemberg, Germany

doi:10.18420/se2025-ws-18

rounding errors that occur during the simulation of quantum circuits can be upper-bounded. Specifically, we improve upon a previously established bound [7] and demonstrate that much tighter bounds are achievable.

Contribution. Our contribution is twofold. We first provide theoretical upper bounds for the errors that can occur during the simulation of quantum circuits only based on the number of gates and the number of qubits. We then show that this bound can be improved to be completely independent of the number of qubits only relying on the size of the largest gate applied in the circuit. Additionally we provide a computable bound which is less tight than the bound presented before but can be compute in floating-point arithmetic without error.

All presented bounds are magnitudes better than the previously known bounds and are suitable to prove absence of relevant rounding errors in a large set of quantum circuits.

Outline. This paper is structured as follows. We start by giving an overview of related work in the relevant fields in section 2. We proceed by introducing some fundamentals and notations in section 3. We then define formally what type of simulation we consider for the remainder of the paper in 4. In section 5 we present our main theoretical contributions on the bounds of errors for quantum simulations. Followed by the practical applications of these bounds to different types of circuits in section 6. We finally mention some future work and conclude in 7.

2 Related work

The challenges of floating-point arithmetic have been extensively examined by researchers from various perspectives. General studies addressing floating-point arithmetic and its associated challenges have been published, such as [6] and [8]. Unlike these works, our approach focuses on a subset of floating-point numbers (e.g., limited to a specific interval) and operations (e.g., base operations), seeking to exploit the unique properties that arise from these restrictions.

In the field of numerical analysis, considerable work has been done on error bounds for matrix multiplications (e.g., [4, 2, 3]). These bounds are generally more applicable to a variety of scenarios but tend to be less refined since they do not leverage the specific properties of particular applications. In contrast, our work presents bounds tailored specifically to quantum simulations.

Surprisingly, floating-point arithmetic has not received much attention in the context of verifying or simulating quantum circuits. Fatima and Markov [5] acknowledge the potential errors introduced by floating-point arithmetic and propose methods to reduce the number of operations required for simulating quantum circuits. Combining their approach with our work could offer a promising direction for future research. Furthermore, Niemann et al. [9] explore the impact of floating-point errors on the compactness of QMDDs. Although this work is similar to ours, it focuses on simulating quantum computations via decision diagrams, while our approach concentrates on matrix-vector multiplication. Thus, despite the similarities, our simulation methodologies differ.

3 Foundations and Notation

We make use of some standard notions and notation in the context of floating-point arithmetic. The maximal relative rounding error, called *unit roundoff*, is $u = 2^{-p}$ where p is the precision (bits of the mantissa). We write $\mathit{fl}\,(x)$ to denote the result of rounding x, i.e., the nearest number to x that is floating-point-representable. That is:

$$x \in \mathbb{R} : |\mathit{fl}\,(x) - x| = \min\{|f - x| : f \in \mathbb{F}\}$$

where $\mathbb{F}$ is the set of floating-point values. Elementwise application of $\mathit{fl}\,(\cdot)$ to vectors and matrices follows naturally. We will also use this notation to indicate that operations are conducted in floating-point arithmetic. That is, $\mathit{fl}\,(a \circ b)$ indicates that a and b are rounded according to $\mathit{fl}\,(\cdot)$, and then the result of the operation $\circ$ is again rounded. We abuse notation to indicate the rounding of arithmetic operations stored in a variable. E.g. if clear from context that $x = a + b$ we use $\mathit{fl}\,(x) = \mathit{fl}\,(a + b)$.

In this paper the absolute value when applied to vectors or matrices is always considered to be applied elementwise unless explicitly stated otherwise.

We repeat standard definitions of vector as well as matrix norms for convenience. For $x \in \mathbb{C}^n$:

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$

$$\|x\|_2 = \left(\sum_{i=1}^{n} |x_i|^2\right)^{\frac{1}{2}}$$

$$\|x\|_\infty = \underset{1 \le i \le n}{max} |x_i|$$

and for $A \in \mathbb{C}^{m\times n}$

$$\|A\|_1 = \max_{1\le j\le n} \sum_{i=1}^{m} |a_{ij}|$$

$$\|A\|_\infty = \max_{1\le i\le m} \sum_{j=1}^{n} |a_{ij}|$$

$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2$$

Since we focus on the simulation of quantum circuits, we primarily consider unitary matrices. A matrix U is unitary if and only if $U^\dagger U = UU^\dagger = I$, where $U^\dagger$ is the complex-conjugate transpose of U. Key properties of unitary matrices relevant to our analysis include [1](Chapter 7, 7.53):

- $\|U\|_2 = 1$,
- Each row and column vector u_i of U satisfies $\|u_i\|_2 = 1$,
- For any element u_{ij} of U, we have $|u_{ij}| \le 1$.

These properties are fundamental in bounding errors within quantum circuit simulations.

4 Simulation of Quantum Circuits

Quantum circuit simulation can leverage various state representations, with the most direct approach modeling states as complex-valued vectors and gates as matrices of corresponding sizes. To maintain clarity and because floating-point operations are particularly susceptible to minor variations, we provide the precise simulation algorithm used in Listing 1. This is taken from our previous paper [7].

This simulation approach operates under a few key assumptions. First, as noted in line 8, it assumes the simulation initiates from a designated state, specifically $|0\rangle$. This is a standard assumption and can be made without loss of generality, as any state can be reached by applying appropriate gates. Second, we assume all gates have dimensions of $n \times n$, where n represents the system's dimensionality. Any smaller gate can be adjusted to match this size without incurring rounding errors, so this assumption also holds without generality loss. Lastly, we assume that all qubits are measured at the circuit's conclusion. While this limits the simulation to algorithms without mid-circuit measurements (or those requiring only partial qubit measurements), it simplifies the analysis in subsequent sections. Nonetheless, the algorithm can be adapted readily to accommodate these scenarios.

All inputs and outputs in this simulation are treated as complex-valued, resulting in a complex-valued state vector. Notably, each complex value could be represented using two real or floating-point numbers with slight adjustments to arithmetic operations. However, to prioritize clarity and readability, we consistently use complex numbers throughout. In this setup, real-valued literals should be interpreted as complex (e.g., 1 is understood as $1 + 0i$).

The algorithm starts by setting the state vector to the initial state $|0\rangle$. This choice is both conventional and practical, ensuring an exact representation of the starting state without rounding errors, which might otherwise arise with an arbitrary initial state. The algorithm then sequentially applies the gates, where each gate application is represented by a matrix multiplication between the current state vector and the gate matrix. Each element of the resulting vector is computed as the dot product of the current state vector with the corresponding row of the gate matrix, performed using classical matrix multiplication. It is essential to keep in mind the operation sequence, particularly the recursive summation in the dot product calculation. Finally, all qubits are measured through a two-step process: (1) calculating the probability of each measurement outcome, and (2) updating the state based on the observed result.

To compute probabilities, the algorithm first calculates the probability of one of the two possible outcomes, with the other determined as its complement. This probability is obtained by summing the squared magnitudes of the relevant state vector elements. In Listing 1, each element is checked to determine if the j-th bit of its index is zero (line 21). Here, the operator // signifies integer division. The squared magnitude of a complex number, as computed in line 22, is simply the sum of the squares of its real and imaginary components.

The second phase, updating the state, involves two main steps. First, elements of the state vector that do not correspond to the observed outcome are set to zero. Then, the state is normalized by dividing the remaining vector by the computed probability, ensuring that the final state vector maintains a unit length. The same condition used to identify elements to zero out can be reused in this step, with the roles of ones and zeros swapped.

The algorithm's final step determines the observed measurement result, which is inherently probabilistic. A random value from the interval $[0, 1)$ is generated and compared to the calculated probability to select the result. In line 25, this comparison establishes which outcome is observed. Two edge cases are handled: when $p = 1$, any random value will trigger the selection of the if-branch, and when $p = 0$, the else-branch is always chosen. This justifies using the half-open interval $[0, 1)$ for the random value, rather than a closed interval.

This completes the description of the quantum circuit simulation algorithm used throughout the paper unless stated otherwise. Alternative simulation methods, like those based on tensor networks, behave quite differently and are not discussed here.

The presented approach to the simulation of quantum circuits represents the standard methodology and is implemented in several quantum simulators (e.g., Qiskit's `BasicSimulator`). However, most simulators incorporate various optimizations aimed at enhancing performance. The impact of these optimizations on the validity of the bounds we derive must be analyzed on a case-by-case basis.

```
Inputs:
    - $A_1, ..., A_i$ (list of n x n matrices)
    - x (vector of length n)
Output:
    - y (vector after application of all gates and measurement of all qubits)
    - $b_1, ..., b_n$ (the observed classical values for all measurements)

y := (1, 0, ... , 0)

for A : gates:
    y' := (0, ...., 0)
    for i = 0 .. n:
        y'[j] := 0
        for j = 0 .. n:
            y'[i] := y[i] + (A[i][j] * x[j])
    y := y'

for j = 0 ... n:
    p := 0
    for i = 0..n:
        if (i // $2^j$) % 2 == 0:
            p := p + $|y[i]|^2$
    p := sqrt(p)
    r := random([0, 1))
    if r < p:
        for i = 0..n:
            if (i // $2^j$) % 2 == 1:
                y[i] := 0
        y := y / p
        $b_j$ := 0
    else:
        for i = 0..n:
            if (i // $2^j$) % 2 == 0:
                y[i] := 0
        y := y / (1 - p)
        $b_j$ := 1
```

Listing 1 Algorithm to simulate a quantum circuit as considered in this paper

5 Improved error bounds

In this section, we present an improved bound for the simulation of quantum circuits, building on a known bound for matrix-vector multiplications as established by Rump in [10]:

$$|Ax - fl(Ax)| \leq n \cdot u \cdot |A| \cdot |x|$$

Here, the inequality is applied element-wise. This shows that the error in a matrix-vector multiplication between a matrix A and a vector x depends on the absolute values of both the matrix and the vector, as well as their dimensions. This bound can be extended directly to matrix norms. We will mainly use the following form:

$$||Ax - fl\,(Ax)\,||_2 \leq n \cdot u \cdot |||A| \cdot |x|||_2$$

This form is derived by applying the norm to both sides of the original inequality. We now demonstrate how this bound can be used to estimate errors in the simulation of quantum circuits. Consider the following notation: Let A_k represent the k-th matrix applied in the quantum circuit, and let x_k denote the quantum state resulting from multiplying this matrix with the previous state vector, i.e., $x_k = A_k x_{k-1}$.

As a first step, we derive a bound for real-valued matrices and vectors, assuming that all A_k matrices are free of rounding errors, i.e., $fl\,(A_k) = A_k$. Additionally, we assume an upper bound b on the norm of each A_k, specifically that $|||A_k|||_2 \leq b$ for all k. Although we will later show that such bounds are always achievable, for now we assume this condition holds. Using these assumptions, we can derive a bound on the error introduced during the simulation of a quantum circuit.

Theorem 5.1. *Given that $A_i \in \mathbb{R}^{n \times n}$ and $fl\,(A_i) = A_i$:*

$$||fl\,(A_i x_{i-1}) - A_i x_{i-1}||_2 \leq (1 + n \cdot u \cdot b)^i - 1$$

or equivalently

$$||fl\,(x_i) - x_i||_2 \leq (1 + n \cdot u \cdot b)^i - 1$$

Remark 1. Note that this bound exhibits exponential scaling in the number of gates; however, we will demonstrate that for a substantial class of circuits, the term $n \cdot u \cdot b$ can be limited to a very small value. Consequently, this exponential growth is less problematic than it may initially appear.

Remark 2. Observe that the number of qubits affects only the matrix size, n. Later, we will establish a bound that is entirely independent of the number of qubits.

Remark 3. Since this bound resembles a typical Wilkinson-type bound, it can be upper-bounded by the more computationally manageable expression $\frac{c \cdot k}{1 - c \cdot k}$, provided $c \cdot k \leq 1$ with $c = n \cdot u \cdot b$.

We continue by proving Theorem 5.1.

Proof. We prove Theorem 5.1 by induction.

Base Case: ($i = 1$)

$$\begin{aligned}
&\|fl(A_1 x_0) - A_1 x_0\|_2 && (1)\\
&\leq n \cdot u \cdot \||A_1||x_0|\|_2 && (2)\\
&\leq n \cdot u \cdot \||A_1|\|_2 \cdot \||x_0|\|_2 && (3)\\
&= n \cdot u \cdot b \cdot 1 && (4)\\
&= (1 + n \cdot u \cdot b)^1 - 1. && (5)
\end{aligned}$$

This derivation primarily relies on definitions introduced above. We use Rump's original bound to get to (2). Notably, since we assume the initial state $x_0 = |1\rangle$, we know $\||x_0|\|_2 = \|x_0\|_2 = 1$, applied in (4). Ultimately, in (5), we confirm that the base case $i = 1$ has the expected form.

Induction Step: Assuming Theorem 5.1 holds for a k, we now show that this implies that it holds for $k + 1$ as well. For brevity, we define $c = n \cdot b \cdot u$.

$$\begin{aligned}
&\|fl(x_{k+1}) - x_{k+1}\|_2 && (1)\\
=&\|fl(A_{k+1} \cdot x_k) - A_{k+1} x_k\|_2 && (2)\\
\leq&\|A_{k+1} \cdot fl(x_k) - A_{k+1} \cdot x_k\|_2 + \|n \cdot u \cdot |A_{k+1}| \cdot |fl(x_k)|\|_2 && (3)\\
\leq&\|A_{k+1}\|_2 \cdot \|fl(x_k) - x_k\|_2 + n \cdot u \cdot b \cdot \|fl(x_k)\|_2 && (4)\\
=&\|fl(x_k) - x_k\|_2 + c \cdot \|fl(x_k) - x_k + x_k\|_2 && (5)\\
\leq&\|fl(x_k) - x_k\|_2 + c \cdot \|fl(x_k) - x_k\|_2 + c \cdot \|x_k\|_2 && (6)\\
=&\|fl(x_k) - x_k\|_2 \cdot (1 + c) + c && (7)\\
\leq&\Big((1 + c)^k - 1\Big) \cdot (1 + c) + c && (8)\\
=&(1 + c)^{k+1} - 1. && (9)
\end{aligned}$$

We discuss these steps in more detail now. We use the original bound to get to (3). Note, however, that the original bound only accounts for the error that is introduced due to the dot-product operation. The errors to compute the two parameters A_{k+1} and x_k are not covered by that. This is why we have to use $fl(x_k)$. Since we assumed $fl(A_k) = A_k$ we can avoid this for the matrix A_k. In (5) we use the fact that $\|A_{k+1}\|_2 = 1$. Note, the difference between $\||A_k|\|_2 \leq b$ and $\|A_k\|_2 = 1$. The same reason allows us to neglect the factor of $\|x_k\|_2 = 1$ in (7). Last but not least we use the induction hypothesis in (8) to replace $\|fl(x_k) - x_k\|_2$ with $(1 + n \cdot u \cdot b)^k - 1 = (1 + c)^k - 1$. Eventually (9) again shows that the error has the expected form and thus concludes the proof. □

Remember that this upper bound applies to a simplified scenario where we consider only real-valued matrices and vectors, with the additional assumption that matrices are exactly representable as floating-point numbers without rounding errors. However, these assumptions are not realistic for quantum computing. Quantum computations typically require complex-valued arithmetic and matrices with elements that are transcendental numbers. Such elements cannot be represented precisely as floating-point values, and any bound derived under these assumptions does not fully account for the intricacies of practical quantum simulations.

We extend our bounds to incorporate rounding errors arising within gate matrices. Specifically, we consider only the rounding errors introduced when each matrix element is approximated to the nearest representable floating-point value. Errors related to the actual computation of each matrix element, however, remain outside the scope of this analysis.

Since each A_i is unitary its elements fall within $[-1, 1]$. Consequently u serves as an upper bound for rounding each element. Even more specifically we can use:

$$|fl(A_i) - A_i| \leq u \cdot |A_i|.$$

Using this idea, we can bound the perturbed matrix norm $fl(A_i)$ as follows:

$$\|fl(A_i)\|_2 \leq \|A_i + u \cdot |A_i|\|_2 \leq 1 + b \cdot u,$$

and similarly,

$$\|fl(|A_i|)\|_2 \leq b + b \cdot u.$$

Using these bounds on $\|fl(A_i)\|_2$ and $\|fl(|A_i|)\|_2$, we obtain:

$$\|fl(x_k) - x_k\|_2 \leq (1 + n \cdot u^2 \cdot b + u \cdot b + n \cdot u \cdot b)^k - 1.$$

This bound is derived from Theorem 5.1, substituting matrix norms with the just derived upper bounds. For brevity, the full induction-based proof is omitted.

This provides an upper bound on the error in quantum circuit simulation given a valid b such that $b \geq \||A_i|\|_2$. The optimal choice under this condition is $b = \sqrt{n}$, justified by the inequality

$$\|A\|_2 \leq \sqrt{\|A\|_1 \cdot \|A\|_\infty}.$$

Given a unitary matrix A and the matrix A' obtained by taking element-wise absolute values of A, each row (or column) v of A' has a 1-norm of 1 due to unitarity. Applying Cauchy-Schwarz to this 1-norm we find:

$$\|v\|_1 = \sum_{i=1}^{n} |v_i| \leq \sqrt{n}\|v\|_2 = \sqrt{n}.$$

It follows that $\|A'\|_1 = \|A'\|_\infty \leq \sqrt{n}$, and thus due to the aforementioned inequality of the norms $\|A'\|_2 \leq \sqrt{n}$, providing the required upper bound for $\||A_i|\|_2$.

5.1 Extending to complex-valued arithmetic

To extend our bounds to complex-valued matrices and vectors, we represent complex arithmetic using real arithmetic by mapping complex numbers to pairs of real numbers. Specifically, any complex vector in $\mathbb{C}^n$ can be transformed into a corresponding real-valued vector in $\mathbb{R}^{2n}$.

Building on this intuition, matrix multiplication in $\mathbb{C}^n$ can also be represented as an equivalent operation in $\mathbb{R}^{2n}$. Given a matrix $A_k \in \mathbb{C}^{n\times n}$ and a vector $x_k \in \mathbb{C}^n$, we define the following equivalent real-valued representations:

$$A'_k = \begin{pmatrix} \mathrm{Re}(A_k) & -\mathrm{Im}(A_k) \\ \mathrm{Im}(A_k) & \mathrm{Re}(A_k) \end{pmatrix} \in \mathbb{R}^{2n\times 2n}$$

$$x'_k = \begin{pmatrix} \mathrm{Re}(x_k) \\ \mathrm{Im}(x_k) \end{pmatrix} \in \mathbb{R}^{2n}$$

Note that if A_k is unitary, then A'_k will also be unitary, and similarly, if x_k is a unit vector, then x'_k will also be a unit vector. This rephrasing of complex matrix multiplication as a real-valued matrix multiplication with doubled dimensions enables us to apply the previously derived bounds for the real-valued case, simply adjusting for the dimension by using $2n$ in place of n throughout. As we set $b = \sqrt{n}$ in the real case this is also affected and we now have to use $b = \sqrt{2n}$.

Consequently, we state our main result:

Theorem 5.2. *For a circuit consisting of k gates $A_i \in \mathbb{C}^{n\times n}$ and initial state $x_0 = |1\rangle$, a simulation as outlined in Fig. 1 (without measurements) yields a final state x_k for which*

$$\begin{aligned} &\|fl(x_k) - x_k\|_2 \\ \leq &(1 + 2 \cdot n \cdot u^2 \cdot b + u \cdot b + 2 \cdot n \cdot u \cdot b)^k - 1 \end{aligned}$$

holds.

This eventually gives us the desired bound for quantum circuit simulations based only on the number of applied gates and qubits. Note, that this bound is not accounting for underflow errors. This could be fixed by adding a small additive term. For the remainder of this paper we will not consider this option but all further extensions could be be adapted accordingly.

5.2 Computable Bound

The derived bound is valid but not directly computable without rounding errors. In most cases where a rough estimate suffices, this may not be a significant issue. However, for formal guarantees, it is necessary to find a bound that can be calculated without incurring rounding errors. A trivially computable bound derived from Theorem 5.2 is:

$$\|fl\,(x_k) - x_k\|_2 \leq fl\left(\frac{8 \cdot n^2 \cdot u \cdot 2^{\lceil \log_2(k) \rceil}}{1 - 8 \cdot n^2 \cdot u \cdot 2^{\lceil \log_2(k) \rceil}} + u\right)$$

given that $\frac{8 \cdot n^2 \cdot u \cdot 2^{\lceil \log_2(k) \rceil}}{1-8 \cdot n^2 \cdot u \cdot 2^{\lceil \log_2(k) \rceil}} < 1$.

We use the fact that the bound in Theorem 5.2 follows a Wilkinson-style form for which the upper bound $\frac{c \cdot k}{1-c \cdot k}$ is known. In our case we have $c = 2 \cdot n \cdot u^2 \cdot b + u \cdot b + 2 \cdot n \cdot u \cdot b$. To make the bound computable, we exploit the error-free nature of multiplication by powers of two in floating-point arithmetic. Since both n and u are powers of two, no adjustments are needed for them. Overapproximating b as $2n$ also yields a power of two and allows us to use $n \cdot u \cdot b$ as an upper bound for both $b \cdot u$ and $n \cdot u^2 \cdot b$ which leaves us with $c = 8 \cdot c \cdot n^2 \cdot u$. Last but not least we overapproximate k by simply calculating the next biggest power of two.

The only rounding error remaining that we have to consider is the division, which is known to yield results in the range $[0, 1]$ (otherwise the condition explicitly given wouldn't hold). Hence, the maximum rounding error is bounded by u, and adding u sufficiently accounts for this error. While this results in a bound that overestimates the error significantly, it remains practical and sufficient for many scenarios (see Sect. 6).

5.3 Considering gate sizes

For now, we have considered the most general case of gates acting on the entire quantum state. While theoretically possible, this is uncommon in practice. In fact, most gates that can be applied on actual quantum devices operate on a maximum of two qubits at a time. Naturally, one would expect that a gate acting on only a small subset of qubits introduces a proportionally smaller error. This expectation holds, and we can use it to tighten our error bound.

#qubits	#gates	old(f)	new(f)	comp(f)	old(d)	new(d)	comp(d)
3	10	1.517e-04	8.107e-05	1.957e-03	1.413e-13	7.550e-14	1.819e-12
3	100	1.669e-02	8.109e-04	1.587e-02	1.554e-11	7.550e-13	1.455e-11
3	10,000	1.686e+02	8.444e-02	-	1.570e-07	7.550e-11	1.863e-09
3	100,000	1.686e+04	1.249e+00	-	1.570e-05	7.550e-10	1.490e-08
5	10	1.032e-03	6.201e-04	3.226e-02	9.609e-13	5.773e-13	2.910e-11
5	100	1.135e-01	6.218e-03	3.333e-01	1.057e-10	5.773e-12	2.328e-10
5	1,000	1.145e+01	6.395e-02	-	1.067e-08	5.773e-11	1.863e-09
5	10,000	1.146e+03	8.587e-01	-	1.068e-06	5.773e-10	2.980e-08
10	100	1.938e+01	2.002e+00	-	1.804e-08	1.029e-09	2.384e-07
10	10,000	-	5.537e+47	-	1.822e-04	1.029e-07	3.052e-05
10	1,000,000	-	-	-	1.823e+00	1.029e-05	1.957e-03
15	10,000	-	-	-	3.293e-02	1.863e-05	3.226e-02
20	10,000	-	-	-	5.960e+00	3.377e-03	-
25	10,000	-	-	-	1.079e+03	8.410e-01	-
30	10,000	-	-	-	-	5.249e+47	-

Tab. 1: Results for the general bound comparing the old bound, the new bound and the computable version of the new bound for single (f) and double (d) precision

#gates	old(f)	new(f)	comp(f)	old(d)	new(d)	comp(d)
10	6.437e-05	3.035e-05	4.886e-04	5.995e-14	2.665e-14	4.549e-13
1,000	7.145e-01	3.039e-03	3.226e-02	6.655e-10	2.665e-12	2.910e-11
100,000	7.152e+03	3.545e-01	-	6.661e-06	2.665e-10	3.725e-09
10,000,000	-	1.510e+13	-	6.661e-02	2.665e-08	4.768e-07
1,000,000,000	-	-	-	6.661e+02	2.665e-06	3.052e-05

Tab. 2: Results for the bound assuming only 1- and 2-qubit gates are applied: comparing the old bound, the new bound and the computable version of the new bound for single (f) and double (d) precision

Since our bound depends explicitly on the size n of the quantum state, which corresponds to the matrix dimensions, we can adjust it straightforwardly. When applying a gate acting on only p qubits, this operation corresponds to a matrix multiplication of size $2^p \times 2^p$. Thus, for any simulation where the largest gate acts on p qubits, we can replace n in our bound with 2^p, yielding a more accurate estimate.

6 Evaluation

In [7], we analyzed the performance of the previously presented bounds in three distinct quantum simulation scenarios. In this section, we revisit these scenarios and compare our new results with those obtained using the bounds from the earlier work. The results are summarized in Tables 1 and 2. Table 1 presents the bounds for the general case, where all gates are assumed to be applied to all qubits. Table 2, on the other hand, lists the bounds under the assumption that only 1- and 2-qubit gates are applied. All calculations for the results in these tables were performed using floating-point arithmetic with double precision ($p = 53$). The bounds are computed based on the theoretical insights provided in the previous section, separately for single and double precision (denoted by the (f) and (d) columns). A dash (−) indicates that the bound could not be computed for a particular combination of qubits and gates, either because the conditions required for the application of the bound were not met or because an overflow occurred during the computation. Note that, since our bounds do not depend on the specific gates used, these findings are applicable to any circuit of the considered size, including well-known quantum algorithms.

Building on these values, we revisit the three quantum circuit simulation scenarios introduced in [7]: the simulation of test circuits, simulations on personal computers (PCs), and high-performance computing (HPC) simulations.

Simulation of Test Circuits In the case of test circuit simulations, where only a small number of qubits (typically up to a handful) and a few dozen gates are considered, we observe that all bounds provide sufficiently accurate results. Even in the worst-case scenario, where single precision is used for the general case with n-qubit gates, the bounds remain relatively small. Notably, for very small circuits, the old bounds outperform the new computable bound, albeit marginally. However, it is surprising that for the general case (when using single precision), even for comparatively small circuits, no meaningful bounds can be derived. This issue is in contrast to the results obtained using double precision, where small circuits do not present any significant challenges in terms of computational accuracy.

Simulation on PCs For simulations that can still be executed on standard desktop PCs, typically involving up to 20-25 qubits, we find that double precision is essential for obtaining meaningful bounds. Even with double precision, however, the bounds become prohibitively large for circuits involving 20 or more qubits. When we consider the more realistic case of only 1- and 2-qubit gates, the newly introduced computable bound is in the order of 10^{-2} for circuits with up to 1000 gates. For the theoretical bound, which is not computable, the results are similarly in the order of 10^{-2} even for circuits with up to 10,000 gates. Once again, double precision provides significantly better bounds, underscoring the importance of higher precision in such simulations.

HPC Simulation For large-scale simulations that can only be carried out on high-performance computing (HPC) systems, the general bound proves to be of limited utility. The bounds for circuits with only 2-qubit gates are also insufficient for this scale of simulation. However, notably, even the computable bound in double precision provides meaningful results (on the order of 10^{-6}) for circuits involving up to 10 million qubits. This highlights the advantage of double precision, particularly in the context of large-scale simulations.

Comparison with Old Bounds When comparing the old bounds presented in [7] with our newly introduced bounds, it is evident that the new bounds provide superior performance in nearly all scenarios, with the exception of very small circuits where the difference is negligible. Even the computable version of the new bound, which is less strong than the theoretical bound, outperforms the old bounds in almost every case. Notably there are some cases in which the an old bound can be provided while the new computable bound is not valid. This is however irrelevant in practice as in these cases both bounds would exceed 1 thus rendering the bound (while theoretically valid) completely meaningless in practice. This demonstrates the effectiveness and applicability of the new bounds, especially for simulations of medium to large-scale quantum circuits.

7 Conclusion and future work

The bounds presented here are well-suited for bounding rounding errors in a broad class of quantum circuits. However, we acknowledge that these bounds—particularly the computable one—are not tight. As a direction for future work, providing a bound that can be proven to be optimal would be a valuable next step. Furthermore, exploring how other simulation methods, such as tensor networks, handle rounding errors would be of great interest. Finally, an important goal is to establish such bounds not only for a naive theoretical simulation but also for practical, realistic simulators.

In summary, we have derived theoretical upper bounds for the errors that may arise in the simulation of quantum circuits due to floating-point arithmetic. The bounds introduced in this paper represent a significant improvement over the current state of the art. Moreover, we have presented a version of these bounds that remains valid even when floating-point arithmetic is used. Overall, we show that both the computable and theoretical bounds are sufficient to prevent significant rounding errors in a wide range of quantum circuits.

References

[1] Sheldon Axler. *Linear algebra done right.* Springer Nature, 2024.

[2] Grey Ballard et al. “Improving the Numerical Stability of Fast Matrix Multiplication”. In: *SIAM Journal on Matrix Analysis and Applications* 37.4 (Jan. 2016), pp. 1382–1418. doi: 10.1137/15M1032168.

[3] James Demmel, Ioana Dumitriu, and Olga Holtz. “Fast Linear Algebra Is Stable”. In: *Numerische Mathematik* 108.1 (Nov. 2007), pp. 59–91. doi: 10.1007/s00211-007-0114-x.

[4] James Demmel et al. “Fast Matrix Multiplication Is Stable”. In: *Numerische Mathematik* 106.2 (Mar. 2007), pp. 199–224. doi: 10.1007/s00211-007-0061-6.

[5] Aneeqa Fatima and Igor L Markov. “Faster schrödinger-style simulation of quantum circuits”. In: *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE. 2021, pp. 194–207.

[6] David Goldberg. “What every computer scientist should know about floating-point arithmetic”. In: *ACM Computing Surveys* 23.1 (1991), pp. 5–48.

[7] Jonas Klamroth and Bernhard Beckert. “Bounding Rounding Errors in the Simulation of Quantum Circuits”. In: *2024 IEEE International Conference on Quantum Software (QSW)*. IEEE. 2024, pp. 99–106.

[8] Jean-Michel Muller et al. *Handbook of floating-point arithmetic*. Springer, 2018.

[9] Philipp Niemann et al. “Overcoming the Tradeoff Between Accuracy and Compactness in Decision Diagrams for Quantum Computation”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39.12 (2020), pp. 4657–4668. DOI: 10.1109/TCAD.2020.2977603.

[10] Siegfried M. Rump. “Error Bounds for Computer Arithmetics”. In: *2019 IEEE 26th Symposium on Computer Arithmetic (ARITH)*. June 2019, pp. 1–14. DOI: 10.1109/ARITH.2019.00011.