Calculation of \( \pi \) on the IBM quantum computer and the accuracy of one-qubit operations

G. A. Bochkin\(^1\) · S. I. Doronin\(^1\) · E. B. Fel’dman\(^1\) · A. I. Zenchuk\(^1\)

Received: 15 January 2020 / Accepted: 10 July 2020 / Published online: 23 July 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

A quantum algorithm for the calculation of \( \pi \) is proposed and implemented on the five-qubit IBM quantum computer with superconducting qubits. We find \( \pi = 3.157 \pm 0.017 \). The error is due to the noise of quantum one-qubit operations and measurements. The results can be used for estimating the errors of the quantum computer and suggest that the errors are purely random.

Keywords Quantum algorithms · Quantum errors · Quantum benchmarking · One-qubit operations

1 Introduction

Many problems were considered for the IBM quantum computer with superconducting qubits [1,2], but high levels of noise preclude solving most of them. The problem of decreasing the noise and increasing the computation accuracy remains very important. Various schemes for solving the noise problem were considered [3–5]. There were also some proposals for the tasks suitable for the existing quantum computers [6–9].

In this paper, we do not try to best classical computers with a quantum one. Instead, we focus on a classical problem that has long been used to highlight the capabilities of classical computers and the power of applied mathematics in the era before computers. The problem consists in computing the number \( \pi \).

Arguably, Archimedes’ results for \( \pi \) from 23 centuries ago have been sufficiently accurate for many applications. Nevertheless, a truly gigantic number of digits have been computed since then. We do not attempt to beat the existing records and focus on getting the best result for \( \pi \) we can extract with the IBM quantum computer.
Our algorithm uses only one-qubit operations. We apply our results to estimate the accuracy of the quantum computer. The precise error rate of an early computer might not be of great interest, but the nature of its errors is. Are the dominant errors random or systematic? Our findings are consistent with purely random errors.

This article is organized as follows: In Sect. 2, the idea of the computational method is presented. The algorithm for calculating \( \pi \) on a quantum computer is given in Sect. 3. We briefly summarize our results in the concluding Sect. 5.

2 The idea

Let us consider a qubit in the initial state \(|0\rangle\). After “rotating” it by \( \varphi \) radians (on the Bloch sphere) around the \( y \)-axis, its state will be \( \exp(-iI_y\varphi)|0\rangle \). (\( \varphi \) is thus the dimensionless time). If it is subsequently measured in the basis \( \{|0\rangle, |1\rangle\} \), then the outcome will be \(|0\rangle \) with the probability \( \cos^2(\varphi/2) = (1 + \cos \varphi)/2 \) and \(|1\rangle \) with the probability

\[
p(\varphi) = \sin^2(\varphi/2) = (1 - \cos \varphi)/2.
\]  

The difference between any two successive roots of the equation \( p(\varphi) = 1/2 \), for example, \( \pi/2 \) and \( 3\pi/2 \), is exactly \( \pi \). Hence, \( \pi \) can be determined from the knowledge of \( p(\varphi) \). Of course, we cannot measure \( p(\varphi) \) exactly. On a real quantum device, we cannot even guarantee that the rotation angle is exactly as desired. We expect, however, that the real rotation angle \( \varphi = ct \), where \( t \) is a value which can be precisely controlled (such as the duration of a control pulse) and \( c \) is a constant. Thus, we can estimate \( p(ct) \) for a given \( t = \varphi/c \) as the fraction \( f(t) \) of the measurements that give the result \(|1\rangle \). To proceed further with calculating \( \pi \), we need to estimate the value of \( c \) as well.

Estimation of the constant \( c \)

Below, we will use the notation \( t_1 = \pi/2c \) and \( t_2 = 3\pi/2c \). These are the first two nonnegative roots of the equation \( p(ct) = 1/2 \). Since

\[
\int_{t_1}^{t_2} \left( p(ct) - \frac{1}{2} \right) \, dt = \frac{1}{c},
\]  

(see Eq. (1)), the area under the experimental curve \( (f(t) - \frac{1}{2}) \) on the interval \([\hat{t}_1, \hat{t}_2]\) (where the hats denote an estimate of the respective variable) provides a way to estimate \( c \).

Due to the experimental imperfections, \(|0\rangle\) and \(|1\rangle\) cannot be obtained with certainty, no matter what operations are performed. For the sake of estimating errors, we approximate the experimental probability of obtaining the result \(|1\rangle \) as

\[
P(t) = \alpha \frac{1 - \cos(ct + \phi_0)}{2} + \beta,
\]  

(3)
with three empirical constants $\alpha$, $\beta$, $\phi_0$ in addition to $c$. That is, $P(t)$ is derived from $p(\varphi)$ by arbitrary linear transforms of both its domain (with parameters $c$ and $\phi_0$) and the image (with parameters $\alpha$, $\beta$). These empirical parameters allow us to account for the initial state not being exactly $|0\rangle$ (it could even be mixed) and measurement errors. In the ideal case $\alpha = 1$, $\beta = \phi_0 = 0$. In any case $\alpha \geq 0$, $0 \leq \beta \leq 1 - \alpha$, since the probability must be between 0 and 1.

3 The algorithm

We assume that the approximate period of $P(t)$ is already known, and time units are chosen so that it is approximately 6. The fraction $f(t)$ will often be called simply “fraction”. Let $T$ be the set of all values of $t$ for which the experiments were performed. The algorithm consists of the following steps:

1. Roughly estimate $\alpha$ and $\beta$ (the measurement imperfections): find $\hat{\alpha}$, $\hat{\beta}$ from the system of equations

   $\begin{align*}
   \min_t f(t) &= \hat{\beta} \\
   \max_t f(t) &= \hat{\alpha} + \hat{\beta}
   \end{align*}$

   (4)

2. Normalize $f(t)$ so that its minimum is 0 and maximum is 1:

   $f_1(t) = \frac{f(t) - \hat{\beta}}{\hat{\alpha}}$

3. Define $\tilde{f}_1(t)$ as the extension of $f_1$ for values of $t$ between $\min T$ and $\max T$ by linear interpolation between neighboring values of $f_1(t)$.

4. Roughly estimate $t_1$ and $t_2$: find two time instants $t$ such that $\tilde{f}_1(t) = 0.5$ using root search methods, $\hat{t}_1$ starting from the point $t = 1.5$ and $\hat{t}_2$ starting from $t = 4.5$.

5. Refine the estimate for $\alpha$ and $\beta$: solve the system

   $\begin{align*}
   \text{mean} \left\{ f_1(t) : t \in T \text{ and } |t - \hat{t}_{\text{minval}}| < \delta \right\} &= \hat{\beta} \\
   \text{mean} \left\{ f_1(t) : t \in T \text{ and } |t - \hat{t}_{\text{maxval}}| < \delta \right\} &= \hat{\alpha} + \hat{\beta}
   \end{align*}$

   (5)

where mean $S$ is the arithmetic mean of the set $S$; $\hat{t}_{\text{maxval}} = \frac{\hat{t}_1 + \hat{t}_2}{2}$ is the estimated maximum point of $P(t)$ calculated from $t_1$ and $t_2$; similarly, $\hat{t}_{\text{minval}} = \frac{3\hat{t}_1 - \hat{t}_2}{2}$ is the estimated minimum point of $P(t)$; $\delta$ is chosen so that the systematic error is much less than the expected random error. (The systematic error is due to $t$ not being the true minimum or maximum. It is bounded from above by $1 - \cos(c\delta)$.) The random error can be quantified by the standard deviation of the Bernoulli distribution divided by $\sqrt{k}$, $\sqrt{\frac{P(t)(1-P(t))}{k}}$, where $k$ is the number of points satisfying $|t - \hat{t}_{\text{minval}}| < \delta$ or $|t - \hat{t}_{\text{maxval}}| < \delta$, and $t = \hat{t}_{\text{minval}} + \delta$ or $t = \hat{t}_{\text{maxval}} + \delta$, respectively.

6. Repeat step 2 with new $\hat{\alpha}$ and $\hat{\beta}$.

7. Refine the estimate for $t_1$ and $t_2$ as follows. For $i = 1, 2$:  

\text{ Springer}
(a) take the experimental times \( t \) in a-neighborhood of \( \hat{t}_i \), that is, \( t \in T \) and \( |t - \hat{t}_i| \leq a \) (we chose \( a = 0.5 \)) and corresponding \( f_1(t) \)

(b) fit a linear function \( \gamma_i t + k_i \) to these \( f_1(t) \) by the least squares method

(c) let the new \( \hat{t}_i \) be the solution to the equation \( \gamma_i \hat{t}_i + k_i = 0.5 \)

8. Find the integral \( I = \int_{\hat{t}_1}^{\hat{t}_2} \left( f(t) - \frac{1}{T} \right) \, dt \) using the trapezoidal rule.

9. Estimate \( \pi \) as \( \frac{\hat{t}_2 - \hat{t}_1}{\hat{t}_2 - \hat{t}_1} \).

In principle, one can attempt to iterate steps 5–7 to improve accuracy. This does not appear to improve accuracy significantly.

The parameters \( c \) and \( \phi_0 \) are related to \( \hat{t}_1 \) and \( \hat{t}_2 \), so despite the fact that we do not explicitly estimate \( c \) and \( \phi_0 \), their estimates can be calculated from \( \hat{t}_1 \) and \( \hat{t}_2 \) if desired.

We wrote code for a five-qubit IBM Q quantum processor (ibmq_5_yourktown-ibmx2). The code specified 8192 measurements on each of the five qubits for nominal rotation angles (corresponding to \( t \) in our algorithm) from 0 to 6.3 with the step of 0.1. The value \( \delta = 0.1 \) (see step 5) was chosen. Each value of \( t \) required a separate job on the quantum platform. The results on qubit #5 (Fig. 1e) contain a large jump near \( t = 4 \), apparently due to a calibration difference between our runs. In the graph for qubit #3 (Fig. 1c), a smaller jump near the time instant \( t = \pi \) can also be observed. The results from those qubits were discarded. The results on the three remaining qubits are well described by Eq. (3) in the region of interest \((1 < t < 5)\). The dependence of the fractions \( f(t) \) on the parameter \( t \) is plotted in Fig. 1. On the three plots corresponding to those qubits, the points are, for the most part, close to the theoretical curve.

Checking the algorithm’s accuracy

In order to check the accuracy of the algorithm, we estimated parameters \( \alpha \) and \( \beta \) from experimental data for the three qubits used in the experiment; generated synthetic data with the probabilities according to Eq. (3) for the same values of \( t \) as in the experiment (see above); ran the algorithm 50 times for each of three pairs \( (\hat{\alpha}, \hat{\beta}) \); and calculated the standard deviation of the result.

We can also use the above method to measure the effect of random errors on any intermediate result produced by the algorithm. We found that the standard deviation of \( (\hat{t}_2 - \hat{t}_1) \) is 0.009 and the standard deviation of \( I \) is 0.006.

The comments on the accuracy of the algorithm below and our conclusion that random errors exceed systematic ones are based on this calculation.

After performing the calculation with the above algorithm and averaging the results from different qubits, we obtained the result \( \pi \approx 3.157 \pm 0.017 \) (0.017 is twice the standard deviation calculated as described above).

4 Discussion

Using more complicated integration rules in step 8 of the algorithm such as Simpson’s rule is pointless in our case, as it will not be more accurate. This can be justified as
follows. We consider the case $c = 1$. Calculating the theoretical value of integral $I$ by the trapezoidal rule with the step of 0.1 gives the value of 0.99917 instead of the exact value $I = 1$. Thus, the error due to the trapezoidal rule is small compared to the error due to limited number of measurements in our case (standard deviation of $I$ in simulations, as stated above, is about 0.006, with the magnitude of measurement errors we observed; it would be 0.0023 in the ideal case of $\alpha = 1$) or the estimation error of $(t_2 - t_1)$.

A possible way to improve the accuracy of the results would be to measure several periods of $p(t)$ instead of half a period, assuming that decoherence is small enough.
5 Conclusions

We suggested a simple quantum algorithm for calculating $\pi$ on a quantum computer and implemented it on the IBM quantum platform. We provide an estimate of the magnitude of random errors and show that our results are consistent with the hypothesis of errors being purely random. The correct value of $\pi$ is within the error bars of the estimated value ($3.157 \pm 0.017$). This accuracy is still behind the celebrated result $3^{10^{-7}} < \pi < 3^{10^{-1}}$ by Archimedes, but it seems that the achievements of classical antiquity are within reach of modern quantum information science.

Acknowledgements This work was performed as a part of a state task (State Registration No. AAAA-A19-11907190017-7). This work is partially supported by the Russian Foundation for Basic Research (Grants Nos. 19-32-80004 and 20-03-00147). The authors are grateful to D. E. Feldman for useful discussion.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Koch, J., Yu, T.M., Gambetta, J., Houck, A.A., Schuster, D.I., Majer, J., Blais, A., Devoret, M.H., Girvin, S.M., Schoelkopf, R.J.: Charge-insensitive qubit design derived from the Cooper pair box. Phys. Rev. A 76, 042319 (2007). https://doi.org/10.1103/PhysRevA.76.042319
2. Chow, J.M., Gambetta, J.M., Magesan, E., Abraham, D.W., Cross, A.W., Johnson, B.R., Masluk, N.A., Ryan, C.A., Smolin, J.A., Srinivasan, S.J., Steffen, M.: Implementing a strand of a scalable fault-tolerant quantum computing fabric. Nat. Commun. 5, 4015 (2014)
3. Temme, K., Bravyi, S., Gambetta, J.M.: Error mitigation for short-depth quantum circuits. Phys. Rev. Lett. 119, 180509 (2017). https://doi.org/10.1103/PhysRevLett.119.180509
4. Kandala, A., Temme, K., Córcoles, A.D., Mezzacapo, A., Chow, J.M., Gambetta, J.M.: Error mitigation extends the computational reach of a noisy quantum processor. Nature 567, 491 (2019)
5. Córcoles, A., Magesan, E., Srinivasan, S.J., Cross, A.W., Steffen, M., Gambetta, J.M., Chow, J.M.: Demonstration of a quantum error detection code using a square lattice of four superconducting qubits. Nat. Commun. 6, 6979 (2015). https://doi.org/10.1038/ncomms7979
6. Zhukov, A., Remizov, S., Pogosov, W., Lozovik, E.: Algorithmic simulation of far-from-equilibrium dynamics using quantum computer. Quantum Inf. Process. 17, 223 (2018)
7. Doronin, S.I., Fel’dman, E.B., Zenchuk, A.I.: Solving systems of linear algebraic equations via unitary transformations on quantum processor of IBM Quantum Experience. Quantum Inf. Process. 19(2), 68 (2020)
8. Kandala, A., Mezzacapo, A., Temme, K., Takita, M., Brink, M., Chow, J.M., Gambetta, J.M.: Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. Nature 549, 242 (2017)
9. Li, Y., Benjamin, S.C.: Efficient variational quantum simulator incorporating active error minimization. Phys. Rev. X 7, 021050 (2017). https://doi.org/10.1103/PhysRevX.7.021050

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.