Mathematical Problems of Computer Science 63, 60–70, 2025.
doi:10.51408/1963-0132

UDC 511

A Quantum Diophantine Equation Solution Finder*

Lara Tatli[1] and Paul Stevenson[2]

[1] University of Durham, Durham, DH1 3LE, UK
[2] University of Surrey, Guildford, Surrey, GU2 7XH, UK
e-mail: lara.tatli@durham.ac.uk, p.stevenson@surrey.ac.uk

Abstract

Diophantine equations are multivariate equations, usually polynomial, in which only integer solutions are admitted. A brute force method for finding solutions would be to systematically substitute possible integer values for the unknown variables and check for equality.

Grover's algorithm is a quantum search algorithm which can find marked indices in a list very efficiently. By treating the indices as the integer variables in the Diophantine equation, Grover's algorithm can be used to find solutions in a brute force way more efficiently than classical methods. We present a hand-coded example for the simplest possible Diophantine equation, and results for a more complicated, but still simulable, equation encoded with a high-level quantum language.

Keywords: Quantum computing, Grover's algorithm, Diophantine equations.
Article info: Received 13 November 2024; sent for review 25 November 2024; accepted 3 April 2025.
Acknowledgements: PDS acknowledges support from UK STFC under grant ST/W006472/1. We acknowledge the use of IBM Quantum Lab in the early stages of this work.

1. Introduction

A Diophantine equation is an equation, typically polynomial, with integer coefficients, in more than one integer variable. A famous example occurs as Fermat's Last Theorem, which states that

$$x^n + y^n = z^n \tag{1}$$

has no solutions for $n \geq 3$ where n, x, y, and z are all natural numbers. The simplest Diophantine equation is linear in two variables and is of the form

$$ax + by = n, \tag{2}$$

*Data Availability: Codes for all parts of this work are available at `https://github.com/LaraTatli18/grovers-algorithm`.

where a, b, and n are given constants. While this equation has well-known solutions, in many other cases, solutions to Diophantine equations are not known (see e.g. the regularly-updated paper by Grechuk keeping track of some open and solved problems [1]). Seeking solutions to Diophantine solutions through numerical search is an established method, where searches can prove the existence of solutions where it is posited that none exist [2].

Here, we bring quantum computing to bear upon the search for Diophantine equation solutions, using Grover's algorithm [3] to look for solutions for the simple linear equation of the form (2). We choose $a = b = 1$ and $n = 5$ arbitrarily for definiteness, and also explore a simple quadratic equation to give an indication of scaling. Both examples are deliberately simple so that they can be encoded in a workable number of qubits on an available simulator. While we are not aware of works explicitly solving Diophantine equations with a quantum search algorithm, we note recent work using Grover's algorithm to perform a series of basic arithmetic procedures through search [4]. In our work we use standard classically-inspired quantum circuits for arithmetic (not using search) and use Grover for the search for equality.

2. Grover's Algorithm as Equation Solution Searcher

We give here a brief discussion of the principles of a quantum search algorithm, following the treatment in Nielsen and Chuang's textbook [5]. The search algorithm generally searches through a search space of N elements. It is supposed that one can work at the level of the index of the elements such that if presented with the index, it is easy to check if it is the element sought. This is the case in our example where checking if given numbers x and y are solutions of the given equation is straightforward by direct substitution and evaluation.

The algorithm uses an oracle, $\mathcal{O}$, which acts as

$$\mathcal{O}|x\rangle|q\rangle \rightarrow |x\rangle|q \oplus f(x)\rangle. \tag{3}$$

Here, $|x\rangle$ is a register of index qubits, and $|q\rangle$ is the oracle qubit. $\oplus$ is addition modulo 2 and $f(x)$ is a function which returns 0 if index x is not a solution to the search problem, and 1 if index x is a solution.

If the oracle qubit is prepared in the state $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$ then the action of the oracle is

$$\mathcal{O}|x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \rightarrow (-1)^{f(x)}|x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right), \tag{4}$$

thus the action of the oracle marks out, with a phase change, components of the register state $|x\rangle$ which are solutions to the problem - i.e. have $f(x) = 1$. The full Grover algorithm then amplifies the states which have been marked, and suppresses the unmarked states, using a "diffuser" circuit. The oracle-diffuser combination together constitute a single Grover iteration. A total of $O(\sqrt{N/M})$ iterations are needed in general to have the solutions selected in the register with high probability, where M is the number of solutions in the N-element space. Note that the standard diffuser requires that valid solutions do not account for the majority of the solution space, but this is the usual condition for an interesting Diophantine equation.

For the case of our linear equation (2), the indexing register works by having $2m$ qubits in which each half encodes one of the numbers x and y. The encoding is made directly in standard binary and we do not consider negative numbers. Clearly the size of m will determine the available integers in the search space, and one must apply ever more qubits to increase the size of the search space, though one benefits from an exponential increase in search space as the number of qubits increases linearly.

For this exploratory study, to find solutions to the equation $x + y = 5$ we use a $2m = 6$ qubit register $|x\rangle$ to encode two 3-bit numbers to add together. The oracle performs the addition and checks the result against the desired solution. The details of the quantum adder we use is given in the next section.

3. Quantum Adder Circuit

A quantum adder capable of calculating the sum of two 3-qubit binary numbers was produced using Qiskit. The adder was designed in such a way that the registers storing the input numbers were not overwritten during the calculation, as is the case with e.g. ripple-carry adders [6]. Retaining the input numbers is useful for use in further calculation, though not vital in our case.

In this setup, shown in Fig. 1, the first 3 qubits, x_0, x_1 and x_2, denote the binary digits representing a natural number x in the format $x_0x_1x_2$, where x_2 is the least significant bit. In the same manner, qubits y_0, y_1 and y_2 denote the natural number y in the format $y_0y_1y_2$. Qubits a_0 and a_1 represent ancillary qubits used to hold carry bits in the addition. Qubits s_0, s_1, s_2 and s_3 denote the solution to $x + y$ in the form $s_0s_1s_2s_3$, where s_3 is the least significant bit. The figure shows all qubits that are needed for the full Grover algorithm. Qubit q_{12} is the oracle qubit $|q\rangle$ as in equation (3).

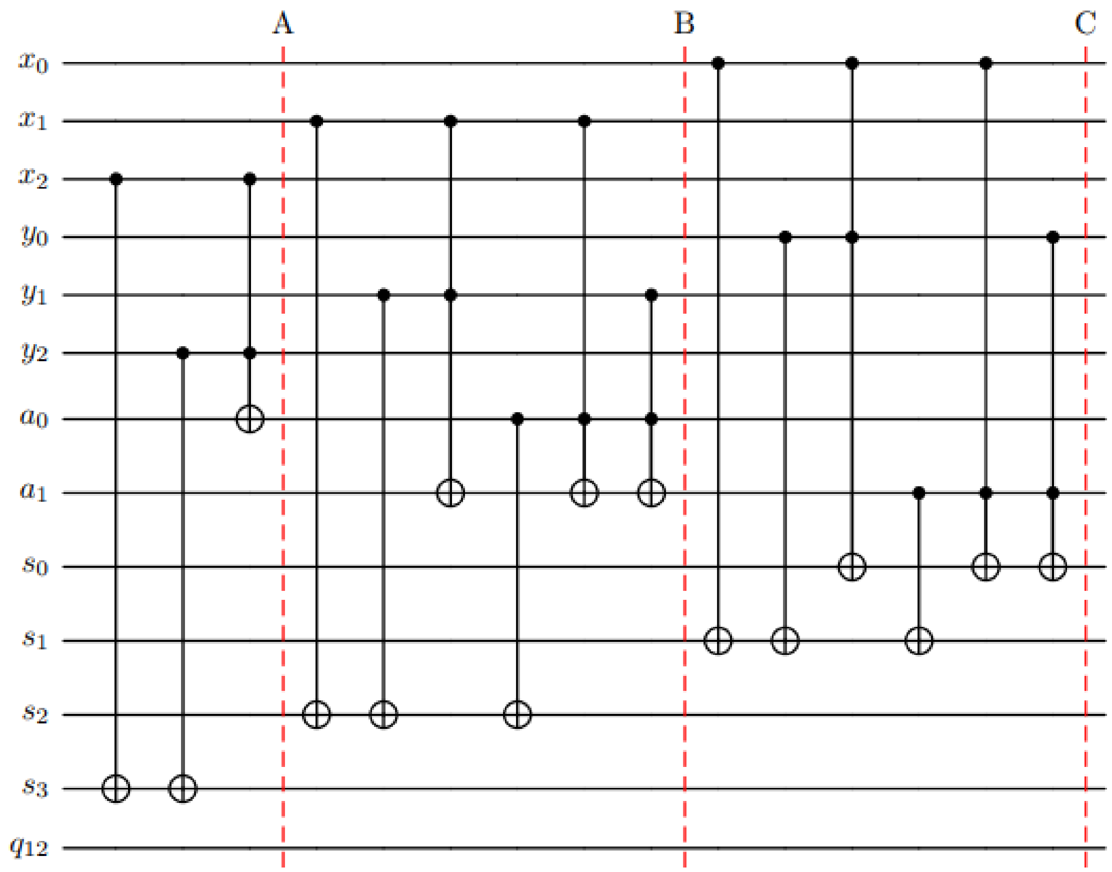

Fig. 1. A diagram of the quantum adder with barriers included to visually indicate each section.

The dividers labelled A, B, and C in the circuit help label different functional parts.

In the section terminated by divider A, an addition operation is performed on the qubits representing the least significant bits x_2 and y_2 using two CNOT gates and one Tofolli gate, with the result stored in the qubit s_3 and the first carry bit stored in a_0.

In the section between dividers A and B, the qubits representing x_1, y_1, and the carry bit a_0 are added using three CNOT gates; the target is set to the sum digit s_2. Three Tofolli gates are used to compute the second carry bit, stored in a_1.

In the section between B and C, the sum digit s_1 is calculated using three CNOT gates acting on the qubits representing x_0, y_0, and the second carry bit in a_1. The final sum digit, s_0, is calculated using three Toffoli gates and takes into consideration the second carry bit.

In total, this adder employs 8 CNOT gates and 7 Toffoli gates collectively acting over 12 qubits. In terms of scaling to larger registers, adding two m-bit numbers requires $4m$ qubits ($2m$ representing the numbers to be added, $m-1$ ancillary carry bits, and $m+1$ to represent the sum). The number of gates is $3m-1$ CNOT gates and $3m-2$ Tofolli gates.

4. Application of Grover's Algorithm

In order to apply Grover's algorithm to solve a linear Diophantine equation $ax+by=n$ in the case $a=b=1$ and $n=5$, it is first necessary to apply a Hadamard gate to each of the qubits $|x_0 \dots x_2, y_0 \dots y_2\rangle$ encoding x and y. This produces the initial superposition state with all possible solution strings present with equal amplitude.

We then construct a quantum oracle capable of "marking" the solutions once queried. This consists of the quantum adder and its inverse circuit with a query circuit in between which applies a phase shift of -1 to the solution qubits of the adder, if and only if, the solution is in the state $|s_0s_1s_2s_3\rangle = |0101\rangle$. All other states are left unchanged. This is achieved using two X-gates and a multi-controlled Toffoli gate targeting q_{12}, configured to be in the $|-\rangle$ state prior to implementing Grover's algorithm. X-gates are re-applied to reverse the computation. The query circuit design used for this example is provided in the left-hand part of Fig. 2.

Each iteration of the oracle is followed by the circuit used for the diffusion operator, which by acting across the six qubits $|x_0 \dots x_2, y_0 \dots y_2\rangle$ amplifies states that sum to give the desired solution only. In this diffuser circuit, shown for our case in the right-hand part of Fig. 2, the combination of Hadamard and X-gates, in conjunction with a multi-controlled Toffoli gate, enable a phase change of -1 to be applied to the initial superposition state. This completes one full iteration of the Grover algorithm. After the desired number of algorithms, one would then perform a measurement on a real quantum computer, identically prepared through many repeated experiments, to build up a histogram of most probable outcomes corresponding to the sought solution(s). The multiple measurements are known as "shots" in the language of quantum computation. In our present example, we simulate our circuit using a full quantum statevector, so present results in the next section by simply reading off the amplitudes of each register state. We show a simulation of a shot-based calcualtion later, for the case of a quadratic equation.

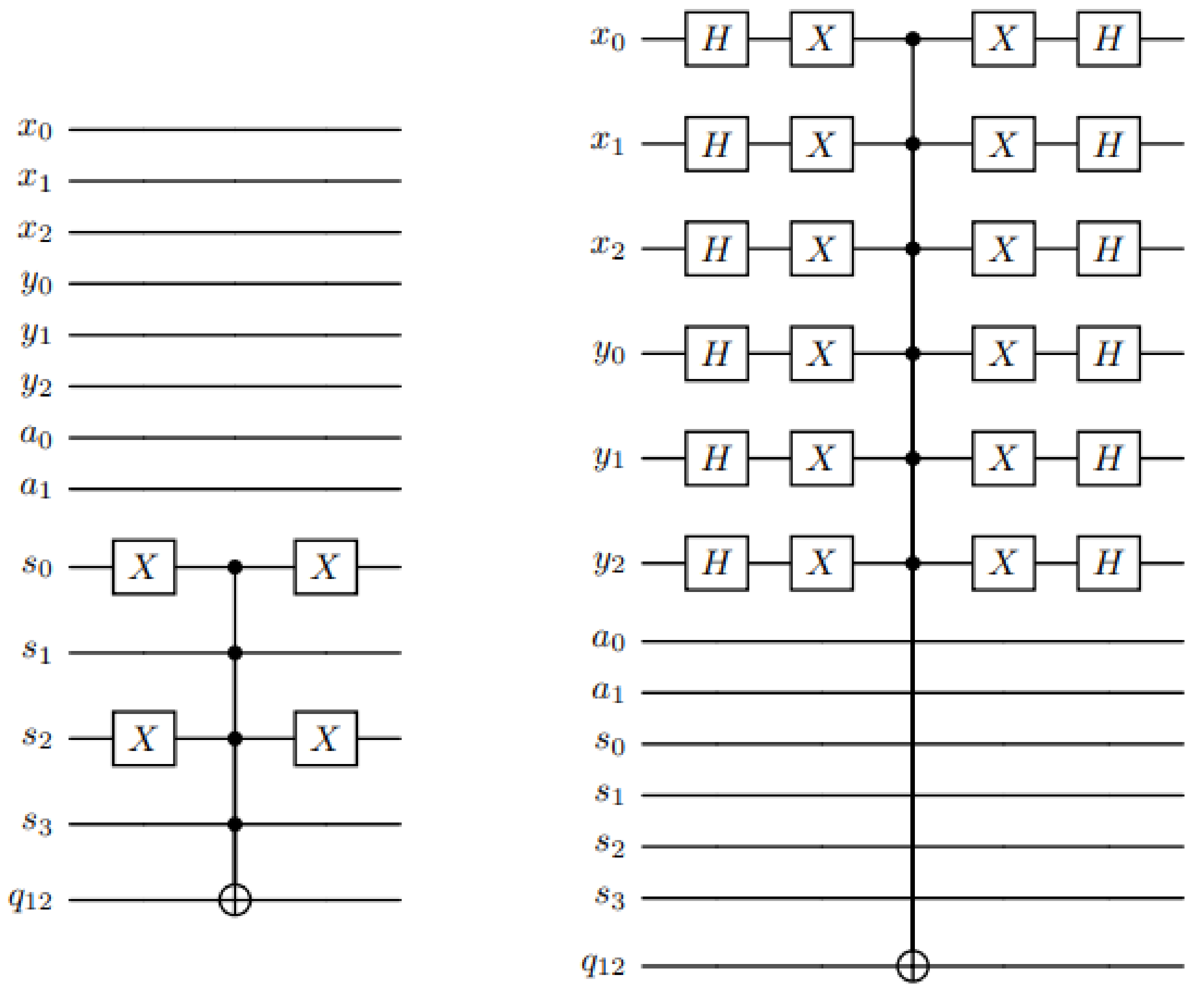

Fig. 2. Left: Diagram of the query circuit and its inverse used for the oracle operation, $\mathcal{O}$, for the case $|s_0 s_1 s_2 s_3\rangle = |0101\rangle$. This circuit is run after the quantum adder circuit and is followed by the inverse quantum adder, forming a complete oracle. Right: The diffuser circuit used to amplify the solution(s).

5. Implementation and Result

The full quantum circuit, including the Hadamards to initialize the superposition of the x and y register qubits and the $|-\rangle$ initialization of the oracle qubit, is shown for one iteration in Fig 3. By running this full quantum circuit on BlueQubit's statevector simulator, it is shown that two iterations of Grover's algorithm are sufficient to generate the full set of solutions to our simple Diophantine equation.

The histogram displayed after one iteration is displayed in Fig. 4; the histogram for two iterations is displayed in Fig. 5. Note that the solution should be read from left to right, with the first three digits representing $x_0x_1x_2$ and the following digits $y_0y_1y_2$.

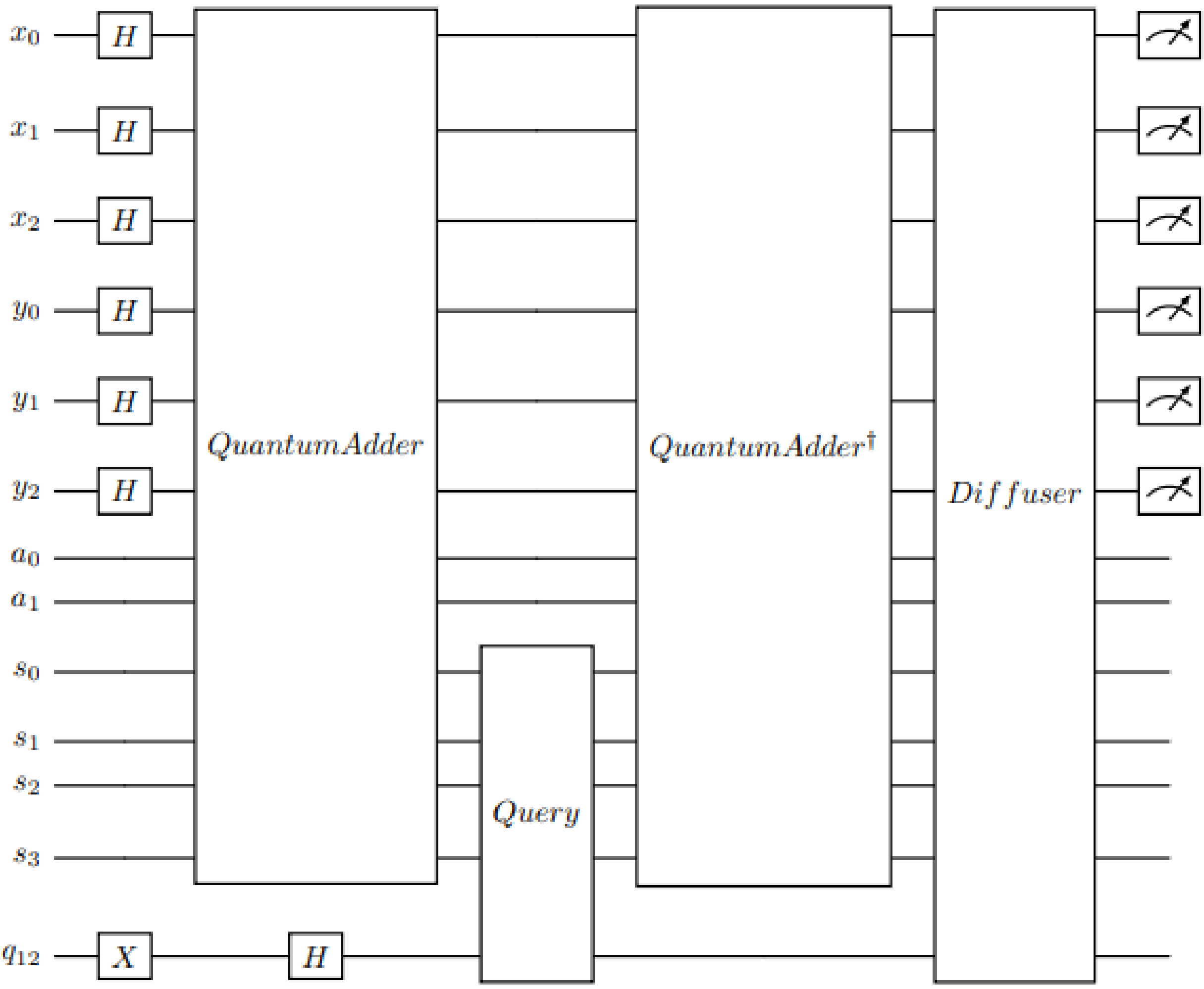

Fig. 3. The complete circuit employing one Grover iteration. The † symbol indicates Hermitian conjugate.

The solutions are seen to be correct solutions of the Diophantine equation $x+y=5$, and we tabulate them for clarity in Table 1.

Table 1. Solution states picked out by Grover's algorithm in search for solutions to Diophantine equation $x+y=5$.

quantum state	x (base 2)	y (base 2)	x (base 10)	y (base 10)	$x+y$ (base 10)
101000	101	000	5	0	5
001100	001	100	1	4	5
011010	011	010	3	2	5
100001	100	001	4	1	5
000101	000	101	0	5	5
010011	010	011	2	3	5

We find that six iterations of Grover's algorithm are required to return to the probability distribution shown in Fig. 4.

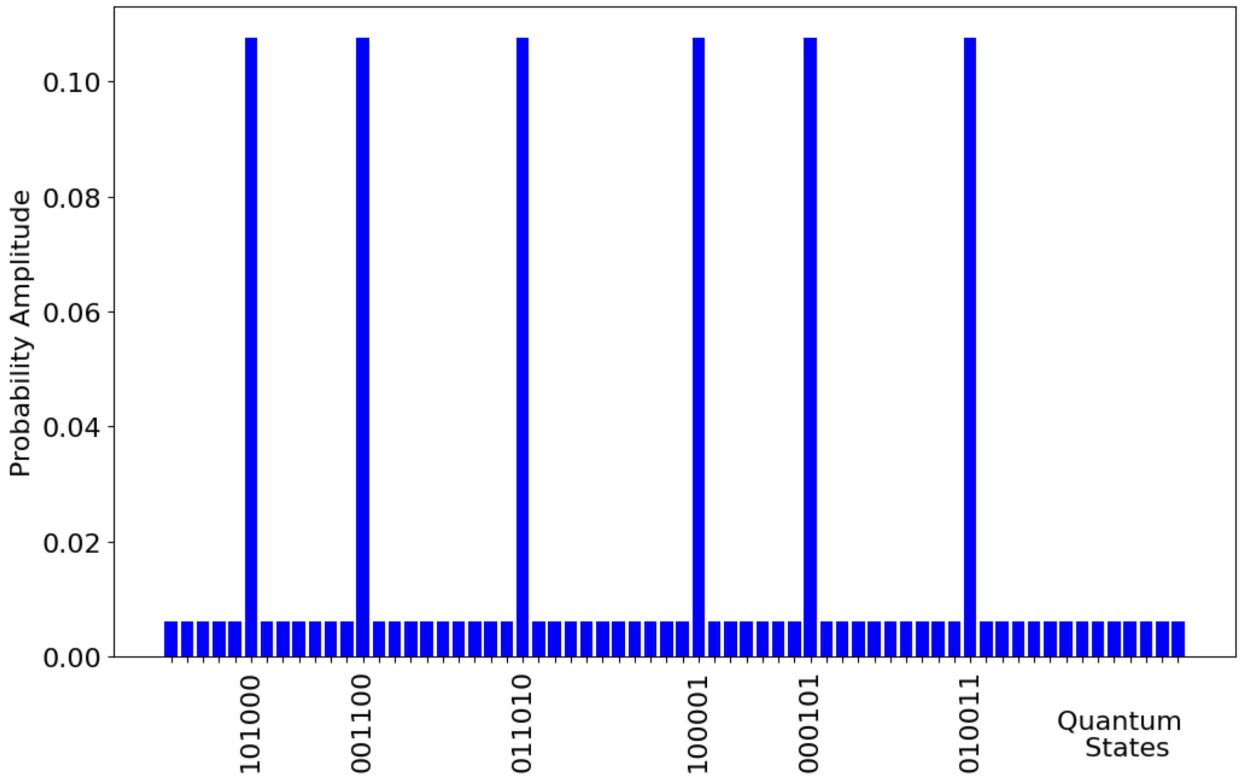

Fig. 4. Histogram for n=1 iterations.

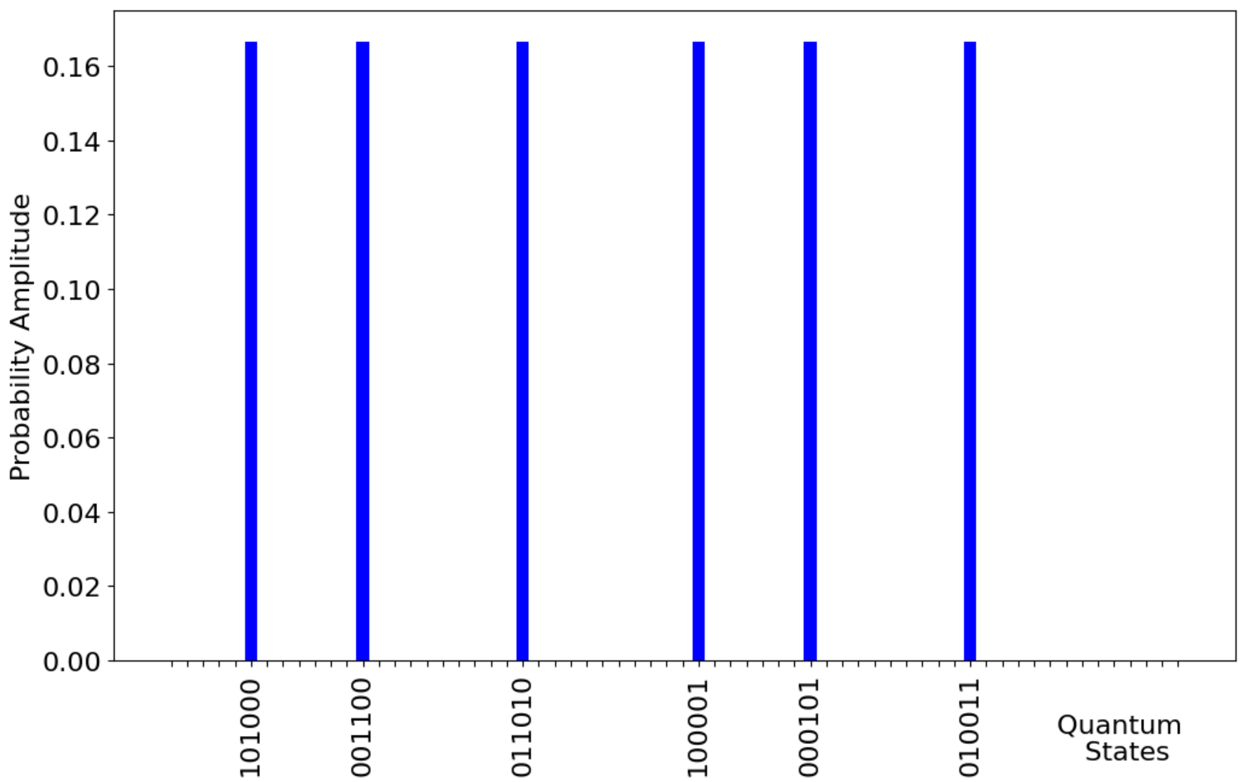

Fig. 5. Histogram for n=2 iterations - probabilities of incorrect solutions effectively become zero.

6. Example with squaring

As an example of a more complicated equation, we look for solutions of the equation

$$x^2 + y^2 = z. \tag{5}$$

The complication of raising variables to a power brings in an increased overhead in ancillary qubits and in depth of quantum circuit necessary to perform the calculations, meaning a more automated method for circuit generation is necessary, as opposed to the hand-made adder used in our first example.

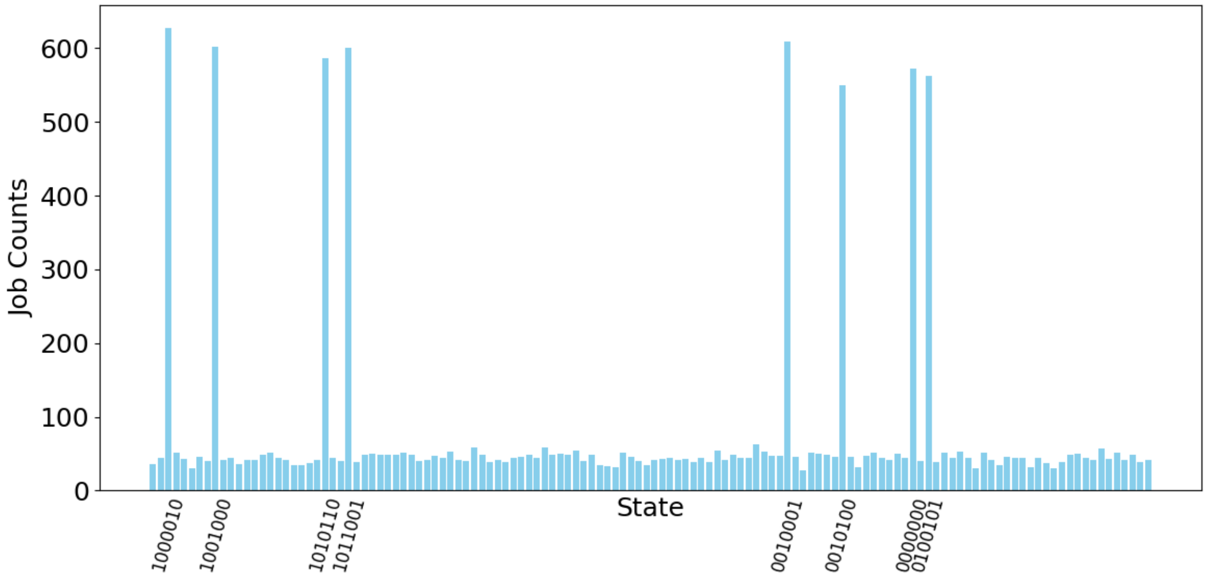

Fig. 6. Grover search for $x^2 + y^2 = z$.

Table 2. Results of simulation of quadratic equation $x^2 + y^2 = z$ using Classiq platform

index	x	y	z
1000010	2	0	4
1001000	0	2	4
1010110	2	1	5
1011001	1	2	5
0010001	1	0	1
0010100	0	1	1
0000000	0	0	0
0100101	1	1	2

We made use of the *Classiq* framework [7], which is able to automate the conversion of quantum algebra into circuit form. The equation (5), when variables x, y, and z are encoded with 2, 2, and 3 qubits respectively, is converted into a 18 qubit circuit with a depth of 502 basis gates. In order to search for Pythagorean triples, the circuit for $x^2 + y^2 = z^2$, with the minimum bit-representation to find the $\{3, 4, 5\}$ triple was designed on the Classiq system, and has a qubit count of 33 and a depth of 981. This latter circuit cannot be simulated on the free Classiq platform and we present results of the simpler equation (5), shown in Fig. 6, using a 10,000 shot simulation, as opposed to the exact statevector calculation for our first example.

The labelled peaks, reading from left to right are shown in Table 2. Note that the encoding used by Classiq is such that the seven bits in the indices encode the variables as $z_0z_1z_2y_0y_1x_0x_1$, witht he least significant bit at the right in each variable encoding. Note that the noisy background for the non-amplified non-solutions in Fig. 6 is due to "shot noise" that comes from the statistical analysis of the quantum measurement.

7. Conclusions

Grover's algorithm can be implemented to search for solutions to simple linear Diophantine equations. We have not attempted implementation on a real quantum computer, and the ability of our circuit to operate on noisy intermediate-scale quantum devices would need to be evaluated. Nevertheless, further work could investigate more complicated Diophantine equations, if access to sufficient real or similated qubits is available. In that case, more interesting unsolved cases, like those listed in Grechuk's paper [1] could be tackled.

Furthermore, we have not attempted to refine or optimize the quantum algorithm, rather concentrating on a straightforward implementation. Techniques to improve the Grover convergence [8] could be applied, while inclusion of a quantum counting approach [9] would allow one to gain knowledge of how many Grover iterations should be applied in advance of performing each calculation. For a more general Diophantine equation solver, such enhancements would be desirable. We also comment that we have preformed a naive brute force search, while standard methods for solving Diophantine equations can be invoked to to reduce the search space.

References

[1] B.Grechuk, "Diophantine equations: a systematic approach", arxiv:2108.08705, 2022. https://doi.org/10.48550/arXiv.2108.08705

[2] R.E. Frye, "Finding $95800^4 + 217519^4 + 414560^4 = 422481^4$ on the connection machine", *Supercomputing 88:Proceedings of the 1988 ACM/IEEE Conference on Supercomputing, Science and Applications*, vol. 2, pp. 1061162, 1988. DOI: 10.1109/SUPERC.1988.74138

[3] L.K. Grover, "A fast quantum mechanical algorithm for database search", *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. STOC 96*, Association for Computing Machinery, New York, NY, USA, pp. 212219, 1996. https://doi.org/10.1145/237814.237866

[4] A. Roy, J.L. Pachuau, G. Krishna and A.K. Saha, "Applying Grovers algorithm to implement various numerical and comparison operations", *Quantum Studies: Mathematics and Foundations*, vol. 11, pp. 291306, 2024. https://doi.org/10.1007/s40509-024-00323-w

[5] M. A. Nielsen and I.L. Chuang, *Quantum Computation and Quantum Information*, 10th Anniversary Edition, Cambridge University Press, Cambridge, UK 2010. https://doi.org/10.1017/CBO978051197666

[6] F. Orts, G. Ortega, E. F. Combarro and E.M. Garzon, "A review on reversible quan tum adders", *Journal of Network and Computer Applications*, vol. 170, 102810, 2020. https://doi.org/10.1016/j.jnca.2020.102810

[7] Classiq: Classiq Platform. https://platform.classiq.io/ Accessed 2024-11-13

[8] I. Abdulrahman, "Enhancing Grovers search algorithm: a modified approach to increase the probability of good states", *The Journal of Supercomputing* vol. 80, pp. 1804818061, 2024. https://doi.org/10.1007/s11227-024-06142-5

[9] S. Aaronson and P. Rall, "Quantum approximate counting, simplified", *Proceedings 2020 Symposium on Simplicity in Algorithms (SOSA)*, Society for Industrial and Applied Mathematics, Philadelphia, PA, pp. 24–32, 2020. https://doi.org/10.51137/1.9781611976014.

Քվանտային Դիոֆանտյան հավասարումների լուծումների որոնիչ

Լարա Թաթլի[1] և Փոլ Սթիվենսոն[2]

[1]Ֆիզիկայի ֆակուլտետ, Դարհեմի համալսարան, Դարհեմ, H1 3LE, Մեծ Բրիտանիա
[2]Մաթեմատիկայի և ֆիզիկայի դպրոց, Սուրեյի համալսարան, Սուրեյ, GU2 7XH, Մեծ Բրիտանիա
e-mail: lara.tatli@durham.ac.uk, p.stevenson@surrey.ac.uk

Ամփոփում

Դիոֆանտյան հավասարումները բազմաչափ հավասարումներ են, սովորաբար բազմանդամ, որոնք ընդունում են միայն ամբողջ թվերի լուծումներ: Լուծումներ գտնելու համար կոպիտ ուժի մեթոդը ներառում է անհայտ փոփոխականների համար հնարավոր ամբողջ թվերի համակարգված փոխարինումը և հավասարության ստուգումը:

Գրովերի ալգորիթմը քվանտային որոնման ալգորիթմ է, որը կարող է շատ արդյունավետորեն գտնել նշված ինդեքսները ցանկում: Դիոֆանտյան հավասարման մեջ ինդեքսները որպես ամբողջ թվերի փոփոխականներ դիտարկելով՝ Գրովերի ալգորիթմը կարող է օգտագործվել կոպիտ ուժի կիրառմամբ լուծումներ գտնելու համար շատ ավելի արդյունավետ, քան դասական մեթոդները: Մենք ներկայացնում ենք Դիոֆանտյան ամենապարզ հնարավոր հավասարման ձեռքով կոդավորված օրինակ և ավելի բարդ, բայց դեռևս մոդելավորելի հավասարման արդյունքներ, որոնք կոդավորված են բարձր մակարդակի քվանտային լեզվով:

Բանալի բառեր՝ քվանտային հաշվարկներ, Գրովերի ալգորիթմ, Դիոֆանտյան հավասարումներ:

Поиск решения квантового диофантового уравнения

Лара Татли[1] и Пол Стивенсон[2]

[1]Кафедра физики, Университет Дарема, Дарем, DH1 3LE, Великобритания
[2]Школа математики и физики, Университет Суррея, Гилфорд,
e-mail: lara.tatli@durham.ac.uk, p.stevenson@surrey.ac.uk

Аннотация

Диофантовы уравнения - это многомерные уравнения, обычно полиномиальные, в которых допускаются только целочисленные решения. Метод грубой силы для поиска решений заключается в систематической подстановке возможных целочисленных значений вместо неизвестных переменных и проверке равенства.

Алгоритм Гровера - это квантовый алгоритм поиска, который может очень эффективно находить отмеченные индексы в списке. Обрабатывая индексы как целочисленные переменные в Диофантовом уравнении, алгоритм Гровера может быть использован для поиска решений грубой силой гораздо эффективнее, чем классические методы. Мы представляем пример с ручным кодированием для простейшего возможного Диофантова уравнения и результаты для более сложного, но все еще моделируемого уравнения, закодированного с помощью квантового языка высокого уровня.

Ключевые слова: квантовые вычисления, алгоритм Гровера, Диофантовы уравнения.