

# Quantum Approach Explains the Need for Expert Knowledge: On the Example of Econometrics

Songsak Sriboonchitta, Hung T. Nguyen, Olga Kosheleva, Vladik Kreinovich, and Thach Ngoc Nguyen

**Abstract** The main purposes of econometrics are: to describe economic phenomena, and to find out how to regulate these phenomena to get the best possible results. There have been many successes in both purposes. Companies and countries actively use econometric models in making economic decisions. However, in spite of all the successes of econometrics, most economically important decisions are not based only on the econometric models – they also take into account expert opinions, and it has been shown that these opinions often drastically improve the resulting decisions. Experts – and not econometricians – are still largely in charge of the world economics. Similarly, in many other areas of human activities, ranging from sports to city planning to teaching, in spite of all the successes of mathematical models, experts are still irreplaceable. But why? In this paper, we explain this phenomenon by taking into account that many complex systems are well described by quantum equations, and in quantum physics, the best computational results are obtained when we allow the system to make kind of imprecise queries – the types that experts ask.

---

Songsak Sriboonchitta  
Faculty of Economics, Chiang Mai University, Thailand  
e-mail: songsakecon@gmail.com

Hung T. Nguyen  
Department of Mathematical Sciences, New Mexico State University  
Las Cruces, New Mexico 88003, USA, and  
Faculty of Economics, Chiang Mai University, Thailand  
e-mail: hunguyen@nmsu.edu

Olga Kosheleva and Vladik Kreinovich  
University of Texas at El Paso, 500 W. University, El Paso, TX 79968, USA  
e-mail: olgak@utep.edu, vladik@utep.edu

Thach Ngoc Nguyen  
Banking University of Ho Chi Minh City, 56 Hoang Dieu 2, Quan Thu Duc, Thu Duc  
Ho Chi Minh City, Vietnam, e-mail: Thachnn@buh.edu.vn

## 1 Formulation of the Problem

**Why aren't we in charge of the world economics?** Since Newton's equations have been discovered, computing a trajectory of a celestial body or of a spaceship became a purely computational problem.

There was a similar hope when the first equations were discovered for describing economic phenomena:

- that mathematical methods would enable us to predict and control economic behavior as we control spaceships,
- that eventually, all the economic problems will be resolved by appropriate computations,
- that eventually, econometricians – researchers who know how to solve the corresponding systems of equations, how to optimize the desired objective function – will be largely in charge of the world economics.

Since then, econometrics has experienced a lot of success stories, but, in spite of all these success stories, we are still not in charge: who is in charge are experts, CEOs, fund managers, bankers, people who may know some mathematical models, but whose main strength is in their expertise – not in knowing these models.

Why? Why are econometricians not in charge of companies – after all, companies are interested in maximizing their profits, so why not let a specialist in maximization be in charge? The fact that this is not happening en masse shows that, in spite of all the successes of econometrics, there is still a big advantage in using expert knowledge.

But why? We do not have an expert with an intuitive understanding of trajectories in charge of computing spaceship trajectories, why is it different in economics?

**Experts are needed not only in economics.** In many others areas of human activity, there is also a surprising need for experts.

For example, in sports, a few decades ago, new sports mathematical methods were developed that drastically improved our understanding of sports phenomena and led to many team successes; see, e.g., [14]. At first, the impression was that the corresponding formulas provide a much better way of selecting team players than the experience of even most experience coaches. However, it soon turned out that relying only on the mathematical models is not a very effective strategy, that much better results can be obtained if we combine the mathematical model with the expert's opinions; see, e.g., [9]. But why?

Same thing with smart cities. Cities often grow rather chaotically, with unintended negative consequences of different decisions, so:

- why not have a computer-based system combining all city services,
- why not optimize the functioning of the city while taking everyone's interests into account?

This seems to be a win-win proposition. This was the original idea behind smart cities. This idea indeed led to many improvements and successes – but it also turned

out that by themselves, the resulting mathematical models do not always provide us with a very good results. Much better results can be obtained if we take expert knowledge into account; see, e.g., [19].

Yet another area where experts are still (surprisingly) needed is teaching. Every time there is a new development in teaching technology, optimistic popular articles predict that these technologies, optimized by using appropriate mathematical models, will eventually replace human teachers. And they don't.

- This was predicted when videotaped lectures appeared.
- this was predicted with current MOOCs – massive open online courses.

And these predictions turn out to be wrong. Definitely, teachers adopt new technologies, these new technologies make teaching more efficient – but attempts to eliminate teachers completely and let an automatic system teach have not yet been successful.

Same with medical doctors: since the very first medicine-oriented expert system MYCIN appeared several decades ago (see, e.g., [3]), enthusiasts have been predicting that eventually, medical doctors will be replaced by expert systems. Definitely, these systems help medical doctors and thus, improve the quality of the health care, but still medical experts are very much in need.

Similar examples can be found in many other areas of human activity. But why are experts so much needed? Why cannot we incorporate their knowledge into automated systems that would thus replace these experts?

**Why cannot we just translate expert knowledge into computer-understandable terms: approaches like fuzzy logic helped, but experts are still needed.** Many researchers recognized the desirability to translate imprecise natural-language expert knowledge into computer-understandable terms. Historically the first successful idea of such a translation was formulated by Lotfi Zadeh under the name of fuzzy logic [23]. This techniques has indeed led to many successful applications; see, e.g., [2, 15, 11, 16, 18]; however, in spite of all these successes, experts are still needed. Why?

**What we do in this paper.** In this paper, we show that this unexpected need for expert knowledge can be explained if we take into account that many complex systems – especially systems related to econometrics and, more generally, with human behavior – are well described by quantum equations [6, 22], equations that were originally invented to describe micro-objects of the physical world. And the experience of designing computers that take quantum effects into account has shows, somewhat unexpectedly, that the best results are attained if instead of asking precise questions, we ask kind of imprecise ones – we will explain this in detail in the following sections.

## 2 Quantum Equations and Quantum Computing: Brief Reminder

**Quantum equations are helpful in econometrics.** Somewhat surprisingly, quantum equations – originally developed for studying small physical objects – have been shown to be useful in describing economic phenomena and, more generally, any phenomena that involves human decision making; see, e.g., [1, 10, 13].

**Let us therefore look for experience of quantum-related decisions.** In view of the above usefulness, when thinking of the best algorithms for making decisions in economics, to look for how decisions are made – and how the corresponding computations are performed – in the quantum world.

**The main idea of quantum computing: a brief reminder.** To perform more and more computations, we need to perform computations faster and faster. In nature, there is a limitation on the speed of all possible physical processes: according to modern physics, all the speeds are bounded by the speed of light –  $c \approx 300\,000$  km/sec. This may sound like a lot, but take into account that for a typical laptop size of 30 cm, the smallest possible time that any signal need to go from one side of the laptop to another is 30 cm divided by  $c$ , which is about 1 nanosecond, i.e.,  $10^{-9}$  seconds. During this nanosecond, a usual several-gigahertz processor – and gigahertz means  $10^9$  operations per second – performs several arithmetic operations. Thus, to make it even faster, we need to make processors even smaller. To fit billions of cells of memory in a small-size computer requires decreasing these cells to the size at which the size of each cell is of almost the same order as the size of a molecule – and thus, quantum effects, i.e., physical effects controlling micro-world, need to be taken into account.

The need to take quantum effects into account when computing was first emphasized by the Nobelist Richard Feynman in his 1982 paper [5]. At first, quantum effects were mainly treated as nuisance. Indeed, one of the features of quantum physics is its probabilistic nature:

- many phenomena cannot be exactly predicted,
- we can only predict the *probabilities* of different outcomes,
- and the probability that a computer will not do what we want makes the computations less reliable.

However, later, it turns out that it is possible, as the saying goes, to make tasty lemonade out of the sour and not-very-edible-by-themselves lemons that life gives us: namely, it turned out that by cleverly arranging the corresponding quantum effects, we can actually speed up computations – and speed them up drastically.

**The main successes of quantum computing: a brief overview.** The first result showing potential benefits of quantum computing was an algorithm developed by Deutsch and Josza ten year after Feynman's paper; see [4] (see also [12] for a pedagogical description of this algorithm). This algorithm solved the following simple-sounding problem:

- given a function  $f(x)$  that transforms one bit (0 or 1) into one bit,
- check whether this function is constant, i.e., whether  $f(0) = f(1)$ .

This may sound like a simple problem not worth spending time on, but it is actually a simple case of a very important practical problem related to high performance computing. In many applications, we have developed software that solves the corresponding system of partial differential equations: it takes as input the initial conditions, the boundary conditions, and produces the results. Solving such systems of equations often requires a lot of computation time; for example:

- accurately predicting tomorrow's weather requires several hours on the fastest modern high performance computer, and
- reasonably accurately predicting where the trajectory of a tornado will go in the next 15 minutes takes even longer than several hours – thus making current predictions practically useless.

One possible way of speeding up computation is based on the fact that:

- while we include all the inputs into our parameters,
- some of current input's bits do not actually affect our results.

This is, by the way, one of the skills that physicists have – in situations like this, figuring out which inputs are important and which can be safely ignored. But even after utilizing all the physicists' expertise, we probably have many bit of data that do not affect tomorrow's weather, i.e., for which, whether we put it bit value 1 or bit value 0 into the corresponding computations, we will get the exact same result:

$$f(\dots, 1, \dots) = f(\dots, 0, \dots).$$

Now we see that the original Deutsch-Josza problem is indeed the simplest case of an important practical problem – important when computing the above simple function  $f(x)$  takes a lot of computation time. If we operate within classical physics, then we have to plug in either 0 or 1 into the given “black box” for computing  $f(x)$ . If we only plug in 0 and not 1, we will know  $f(0)$  but not  $f(1)$  – and thus, we will not be able to know whether the values  $f(0)$  and  $f(1)$  are the same. To check whether the given function  $f(x)$  is a constant, we therefore need to call the function  $f(x)$  *two* times.

An interesting result of Deutsch and Josza is that in quantum computing, we can find the answer by using only *one* call to the function  $f(x)$  – in the next section, we will explain how this is possible and how this is related to the need for expert knowledge.

This result opened the floodgates for many other efficient quantum algorithms. One of the first was Grover's algorithm for a fast search in an unsorted array [7, 8]. The search problem is becoming more and more important every day, with the increasing amount of data coming in. Ideally, we should sort all this data – e.g., in alphabetic order – and thus make it easier to search, but in practice, we often no time for such sorting, and thus, store the data in a non-sorted order, in memory cells

$$c_1, c_2, \dots, c_n.$$

Suppose now that we want to find a record  $r$  in this database. For example, suppose that an act of terror has happened, the surveillance system recorded the faces of penetrators, and to help stop further attacks, we want to find if these faces have appeared in any of the previously recorded surveillance video recordings.

A natural way to find the desired record is to look at all  $n$  stored records one by one until we find a one. In this process, if we look at fewer than  $n$  records, because we may thus miss the record  $c_i$  containing the desired information. Thus, in the worst case, to find the desired record, we must spend time  $c \cdot n = O(n)$ , where  $c$  is the average time needed to look into a single record – and within classical physics, no faster algorithm is possible. Interestingly, Grover’s quantum algorithm searches for the record much faster – in time proportional to  $\sqrt{n}$ .

There are many other known effective quantum algorithms. The most well known is Shor’s fast factorization algorithm [20, 21] that enables us to factorize large integers fast. This sounds like an academic problem until one realizes that most computer encryption that we use know – utilizing the so-called RSA algorithm – is based on the difficulty of factorizing large integers. So, if Shor’s algorithm becomes practical, we will be able to read all the encrypted messages that have been sent so far – this is why governments and companies all over the world try to implement this algorithm.

*Comments.*

- Shor’s result would not mean, by the way, that with the implementation of quantum computing, encryption will be impossible – researchers have invented unbreakable quantum encryption algorithm which are, by the way, already used to convey important messages. This algorithm and many other quantum computing algorithms can be found in [17].
- In the following section, we will briefly mention how exactly quantum computers achieve their speedup – and how this is related to the need for experts.

### 3 How Quantum Computers Achieve Their Speedup and How This Explains the Need for Imprecise Expert Knowledge

**Superposition: a specific feature of quantum world.** One of the important specific features of the quantum world is that in addition to classical (non-quantum) states, we can have linear combinations (called *superpositions*) of these states. This is a very non-intuitive notion, this is one of the reasons why Einstein was objecting to quantum physics: for example, how can one imagine a superposition of a live cat and a dead cat? Intuitive or not, quantum physics has been experimentally confirmed – while many more intuitive alternative theories ended up being rejected by the experiments.

Let us thus illustrate this idea on the example of quantum states of a bit (a quantum bit is also called a *qubit*). In non-quantum physics, a bit has two states: 0 and 1. In quantum physics, these states are usually denoted by  $|0\rangle$  and  $|1\rangle$ .

In quantum physics, in addition to the two classical states  $|0\rangle$  and  $|1\rangle$ , we also allow superpositions, i.e., states of the type  $c_0 \cdot |0\rangle + c_1 \cdot |1\rangle$ , where  $c_0$  and  $c_1$  are complex numbers. The meaning of this state is that when we read the contents of this bit – i.e., if we try to measure whether we will get 0 or 1:

- we will get 0 with probability  $|c_0|^2$ , and
- we will get 1 with probability  $|c_1|^2$ .

Since we will always find either 0 or 1, these two probabilities must add up to 1:  $|c_0|^2 + |c_1|^2 = 1$ . This is the condition under which the above superposition is physically possible.

**How superpositions are used in Deutsch-Josza algorithm.** In the quantum world, superpositions are “first-class citizens” in the sense that:

- whatever one can do with classical states, we can do with superpositions as well.

In particular:

- just like we can use 0 and 1 as inputs to the algorithm  $f(x)$ ,
- we can also use a superposition as the corresponding input.

And this is exactly the main trick behind the Deutsch-Josza algorithm: that instead of using the classical state (0 or 1) as an input, we use, as the input, a superposition state

$$\frac{1}{\sqrt{2}} \cdot |0\rangle + \frac{1}{\sqrt{2}} \cdot |1\rangle,$$

a state in which we can get 0 or 1 with equal probability

$$\left| \frac{1}{\sqrt{2}} \right|^2 = \frac{1}{2}.$$

**How superpositions are used in Grover’s algorithm.** In the non-quantum approach, all we can do is select an index  $i$  and ask the system to check whether the  $i$ -th record contains the desired information. In contrast, in quantum mechanics, in addition to submitting an integer  $i$  as an input to the database, we can also submit a superposition of different indices:

$$c_1 \cdot |1\rangle + c_2 \cdot |2\rangle + \dots + c_i \cdot |i\rangle + \dots + c_n \cdot |n\rangle,$$

as long as this superposition is physically meaningful, i.e., as long as all the corresponding probabilities add up to 1:

$$|c_1|^2 + |c_2|^2 + \dots + |c_i|^2 + \dots + |c_n|^2 = 1.$$

This is exactly how Grover’s algorithm achieves its speedup – by having such superpositions as queries.

*Comment.* A similar idea underlies Shor’s fast factorization algorithm. Namely, a usual way to factorize a large number  $N$  is to try all possible prime factors  $p \leq \sqrt{N}$ .

In Shor’s algorithm, crudely speaking, instead of inputting a single prime number  $p$  into the corresponding divisibility-checking algorithm, we input an appropriate superposition of the states  $|p\rangle$  corresponding to different prime numbers.

**How all this implies the need for experts.** How can we interpret a superposition input in commonsense terms? For example, in the search-in-the-database problem:

- A traditional query would be to select an index  $i$  and to check whether the  $i$ -th record contains the desired information.
- In quantum computing, we do not select a specific index  $i$ , the query may affect several different indices with different probabilities.

This is exactly the same effect as when an expert asks something like “is one of the earlier records containing the desired information” – meaning maybe record No. 1, maybe record No. 2, etc. Of course, the result of this query is also probabilistic (imprecise): we do not get the exact answer to this question, we get an imprecise answer – which would correspond to something like “possibly”.

In other words, queries like the ones asked by quantum algorithms are very similar to imprecise queries that real experts make. The main lesson of quantum computing is thus that:

- normally, when we start with such imprecise queries, we try to make them more precise (“precisiate” them, to use Zadeh’s term from fuzzy loigic), while
- quantum computing shows that in many important cases, *it is computationally more beneficial to ask such imprecise queries than to ask precise ones.*

In other words, quantum computing proves that combining precise computations with imprecise expert-type reasoning is often beneficial – which explains the somewhat surprising empirical need for such expert reasoning.

## Acknowledgments

This work was supported by the Center of Excellence in Econometrics, Faculty of Economics, Chiang Mai University, Thailand. We also acknowledge the partial support of the US National Science Foundation via grant HRD-1242122 (Cyber-ShARE Center of Excellence).

## References

1. B. E. Baaquie, *Quantum Finance: Path Integrals and Hamiltonians for Options and Interest Rates*, Cambridge University Press, New York, 2004.
2. R. Belohlavek, J. W. Dauben, and G. J. Klir, *Fuzzy Logic and Mathematics: A Historical Perspective*, Oxford University Press, New York, 2017.
3. B. G. Buchanan and E. H. Shortliffe, *Rule Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Addison-Wesley, Reading, Massachusetts, 1984.

4. D. Deutsch and R. Jozsa, "Rapid solutions of problems by quantum computation", *Proceedings of the Royal Society of London, Ser. A*, 1992, Vol. 439, pp. 553–558.
5. R. P. Feynman, "Simulating physics with computers", *International Journal of Theoretical Physics*, 1982, Vol. 21, No. 6/7, pp. 467–488.
6. R. Feynman, R. Leighton, and M. Sands, *The Feynman Lectures on Physics*, Addison Wesley, Boston, Massachusetts, 2005.
7. L. K. Grover, "A fast quantum mechanical algorithm for database search", *Proceedings of the 28th ACM Symposium on Theory of Computing*, 1996, pp. 212–219.
8. L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack", *Physical Reviews Letters*, 1997, Vol. 79, No. 2, pp. 325–328.
9. T. S. Grover with S. L. Wenk, *Relentless: From Good to Great to Unstoppable*, Scribner, New York, 2014.
10. E. Haven and A. Khrennikov, *Quantum Social Science*, Cambridge University Press, Cambridge, UK, 2013.
11. G. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, Upper Saddle River, New Jersey, 1995.
12. O. Kosheleva and V. Kreinovich, "How to Introduce Technical Details of Quantum Computing in a Theory of Computation Class: Using the Basic Case of the Deutsch-Jozsa Algorithm", *International Journal of Computing and Optimization*, 2016, Vol. 3, No. 1, pp. 83–91.
13. V. Kreinovich, H. T. Nguyen, and S. Sriboonchitta, "Quantum Ideas in Economics Beyond Quantum Econometrics", In: Ly H. Anh, Le Si Dong, V. kreinovich, and Nguyen Ngoc Thach (eds.), *Econometrics for Financial Applications*, Springer Verlag, Cham, Switzerland, 2018, pp. 146–151.
14. M. Lewis, *Moneyball: The Art of Winning an Unfair Game*, W. W. Norton, New York, 2004.
15. J. M. Mendel, *Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions*, Springer, Cham, Switzerland, 2017.
16. H. T. Nguyen and E. A. Walker, *A First Course in Fuzzy Logic*, Chapman and Hall/CRC, Boca Raton, Florida, 2006.
17. M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, 2000.
18. V. Novák, I. Perfilieva, and J. Močkoř, *Mathematical Principles of Fuzzy Logic*, Kluwer, Boston, Dordrecht, 1999.
19. K. Schehtner, "Bridging the adoption gap for smart city technologies: an interview with Rob Kitchin", *IEEE Pervasive Computing*, 2017, Vol. 16, No. 2, pp. 72–75.
20. P. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer", *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, Santa Fe, New Mexico, November 20–22, 1994.
21. P. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer", *SIAM J. Sci. Statist. Comput.*, 1997, Vol. 26, pp. 1484–ff.
22. K. S. Thorne and R. D. Blandford, *Modern Classical Physics: Optics, Fluids, Plasmas, Elasticity, Relativity, and Statistical Physics*, Princeton University Press, Princeton, New Jersey, 2017.
23. L. A. Zadeh, "Fuzzy sets", *Information and Control*, 1965, Vol. 8, pp. 338–353.