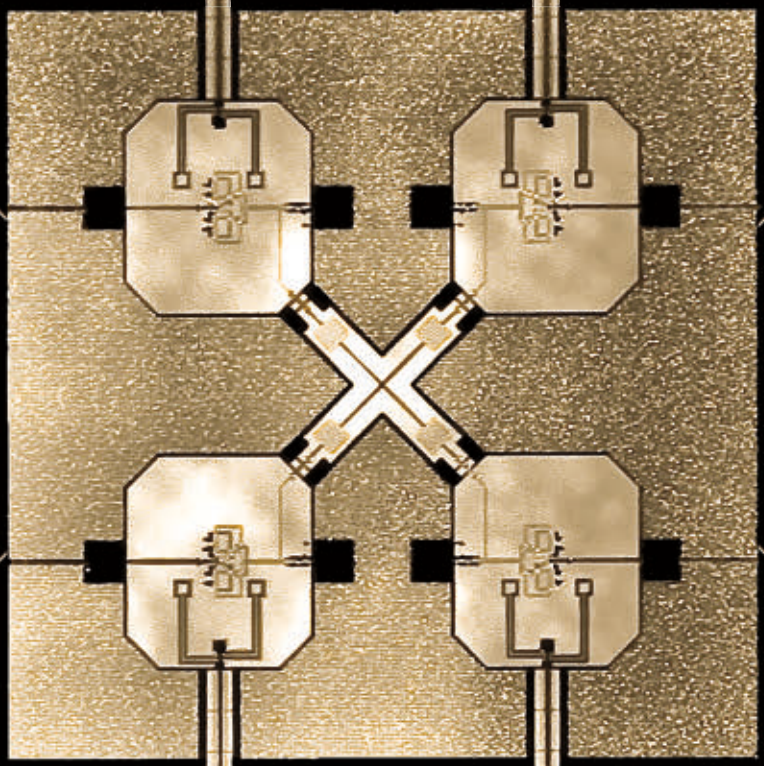
Crossroads The ACM Magazine for Students **FALL 2016** VOL.23 · NO.1 XRDS.ACM.ORG

XRDS



Quantum Computing

Black Holes, Quantum Mechanics, and the Limits of Polynomial-time Computability

Quantum Algorithms for Machine Learning

Association for Computing Machinery

Come join us at CSCW 2017!

CSCW 2017

PORTLAND, OR

DoubleTree by Hilton Hotel

February 25 - March 1, 2017

For more information about the conference and to register, please visit: cscw.acm.org/2017/

Facebook: www.facebook.com/acmCSCW/
Twitter: @ACM_CSCW
Sina Weibo: ACM_CSCW (in Chinese)

The ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW) is the premier venue for presenting research in the design and use of technologies that affect groups, organizations, communities, and networks. Bringing together top researchers and practitioners from academia and industry, CSCW explores the technical, social, material, and theoretical challenges of designing technology to support collaborative work and life activities.

The scope of CSCW spans socio-technical domains including work, home, education, healthcare, the arts, leisure, and entertainment. The conference seeks novel research results or new ways of thinking about, studying, or supporting shared activities in these and related areas.

Upcoming Deadlines

Workshop Proposals: *Oct. 17, 2016*

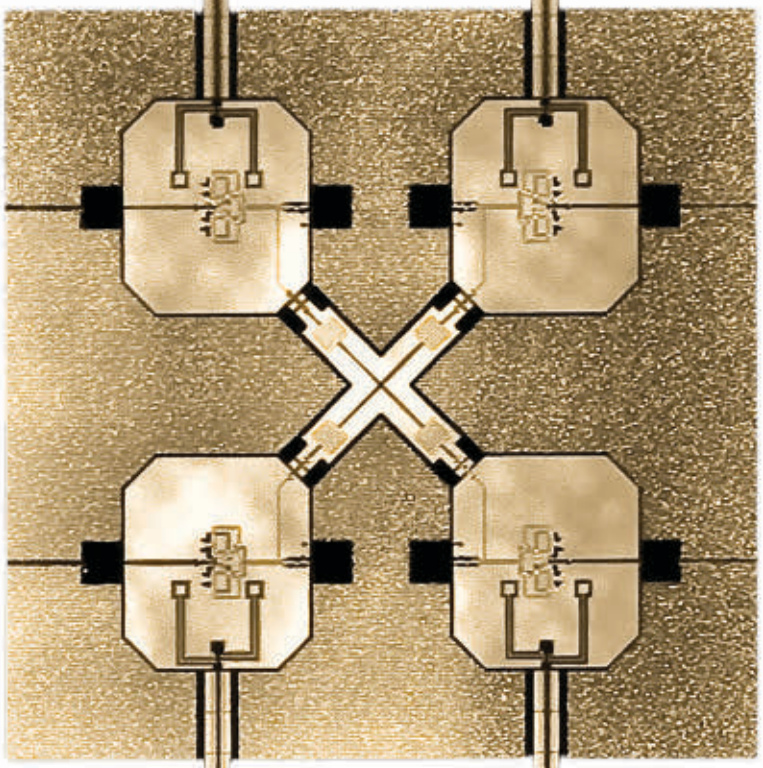
Interactive Posters: *Nov. 4, 2016*

Panels: *Nov. 4, 2016*

Doctoral Colloquium: *Nov. 4, 2016*

Demonstrations: *Nov. 4, 2016*

XRDS
Crossroads
The ACM Magazine for Students
FALL 2016 VOL.23 · NO.1



Cover Image by Erik Lucero, Courtesy of Martinis Group/UCSB

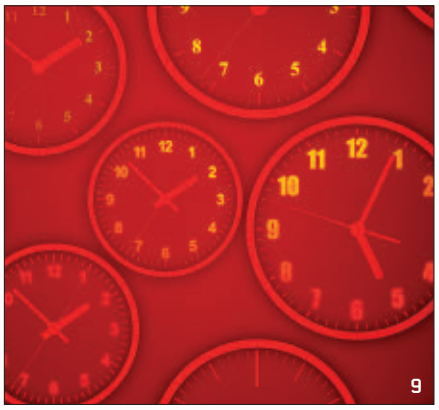


Image by Katykin/Shutterstock.com

begin

5 **LETTER FROM THE EDITORS**

7 **INIT**
Quantum Computation: Double majoring in physics and computer science
By Dawei Ding

9 **ADVICE**
Time Management as a Ph.D.
By Andrew J. Hunsucker

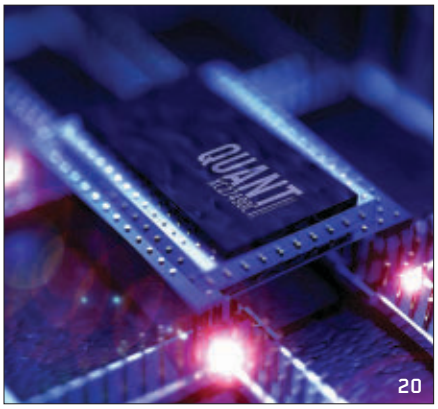
11 **CAREERS**
One Thousand Interviews
By Geerten Peek and Ahmet Taspinar

13 **BLOGS**
CHI 2016: Global, Diverse, Good
By Nur Al-huda Hamdan

An Introduction to Gamification in Human-Computer Interaction
By Gustavo Fortes Tondello

18 **UPDATES**
"ANN" Helps Mario Rescue Princess Toadstool
By Daniel López Sánchez

19 **MILESTONES**
Quantum Milestones
By Jay Patel



features

20 **FEATURE**
Quantum Algorithms for Machine Learning
By Bingjie Wang

25 **FEATURE**
Many-body Quantum Mechanics: Too big to fail?
By Michael L. Wall, Arghavan Safavi-Naini, and Martin Gärttner

30 **FEATURE**
Black Holes, Quantum Mechanics, and the Limits of Polynomial-Time Computability
By Stephen P. Jordan

34 **FEATURE**
Reliable Quantum Circuits Have Defects
By Alexandru Paler, Austin G. Fowler, and Robert Wille

40 **FEATURE**
Establishing Quantum Advantage
By Adam Bouland

45 **FEATURE**
Programming Quantum Computers Using 3-D Puzzles, Coffee Cups, and Doughnuts
By Simon J. Devitt

52 **FEATURE**
Black Holes and the Limits of Quantum Information Processing
By Brian Swingle

57 **FEATURE**
Undecidability of the Spectral Gap
By Johannes Bausch

62 **PROFILE**
David Deutsch: Understanding computation as a consequence of physics
By Adrian Scoică

end

64 **LABZ**
UC Berkeley's Quantum Computing Group
By Seung Woo Shin

65 **BACK**
The RSA Trap
By Asmaa Rabie

66 **HELLO WORLD**
The Infinite Mixtures of Food Products
By Marinka Zitnik

68 **ACRONYMS**

68 **POINTERS**

70 **EVENTS**

72 **BEMUSEMENT**

Association for Computing Machinery

Left Image by Welcomia/Shutterstock.com; Middle Image by Agsandrew/Shutterstock.com; Right Image by Carol VanDyke

XRDS

Association for Computing Machinery

SUBSCRIBE

Subscriptions ($19 per year includes *XRDS* electronic subscription) are available by becoming an ACM Student Member www.acm.org/membership/student

Non-member subscriptions: $80 per year http://store.acm.org/acmstore

ACM Member Services
To renew your ACM membership or *XRDS* subscription, please send a letter with your name, address, member number and payment to:

ACM General Post Office
P.O. Box 30777
New York, NY
10087-0777 USA

Postal Information
XRDS (ISSN# 1528-4981) is published quarterly in spring, winter, summer and fall by Association for Computing Machinery, 2 Penn Plaza, Suite 701, New York, NY 10121. Application to mail at Periodical Postage rates is paid at New York, NY and additional mailing offices.

POSTMASTER: Send addresses change to: *XRDS: Crossroads*, Association for Computing Machinery, 2 Penn Plaza, Suite 701, New York, NY 10121.

Offering# XRDS0171
ISSN# 1528-4972 (print)
ISSN# 1528-4980 (electronic)

From Prototype to Product: Deployment strategies in computer science research

Here's a scenario computer science students may encounter: You've finished building a prototype, performed a small user study, and perhaps published a paper. Now you'd like to find a way to get what you've built into the hands of more people. In short, you're interested in deployment. There are many benefits in deploying your technology to a broader audience. Observing large numbers of people using your work can help you focus on the problem you're targeting, or quickly alert you to weaknesses. For those who are engaged in academic research, in-the-wild studies can help you understand how your technology performs in an ecologically valid context. Other people may use your system in unexpected ways or connect it to a new domain, which can fuel future innovation. In addition to satisfying intellectual curiosity, there is fundamental value in using computer science research to produce useful technology.

There are many different pathways to deployment, each with different limitations and benefits. It can be difficult to navigate the transition from a prototype to product, and it's often unclear where to start. But here are a few possible starting points for getting your technology out in the world.

One approach to deploying research-based technology is to incorporate it into an existing system, a process known as "tech transfer." Frequently tech transfer occurs in industrial development, but it's also possible in academic or non-profit contexts. An example is the Scratch online community. Started by Andres Monroy-Hernández, a former student in the Lifelong Kindergarten group at the MIT Media Lab, the Scratch community began as an experimental online platform for young people to upload their projects from the desktop-based Scratch programming environment. Monroy-Hernández and fellow researchers used the community to better understand how young people were sharing and collaborating with one another. As the community grew, it gradually became integrated with the Scratch programming environment. Eventually, the programming environment and the community merged into an online application. After Monroy-Hernandez graduated, Scratch continued to grow and evolve. Today, the online community is managed by several full-time staff, and currently hosts more than 15 million shared projects.

Scratch is a great example of how tech transfer enables you to build on the resources of an established technology, making it possible to rapidly scale your impact. There are tradeoffs, however, to integrating your technology into an existing system. Frequently, tech transfers involve navigating internal company politics and strict licensing requirements. Core aspects of your design may also have to be significantly modified or removed altogether. If you end up being involved in the actual transfer process, you can learn a lot about productization

engineering, but it can also be incredibly time consuming. Furthermore, while productization doesn't require developing new features, engineering a technology for wide release is often the most technically challenging aspect of a project. It's difficult to pursue a tech transfer without a pre-existing relationship with the company or organization involved. Tech transfer is easier if you're working in an industry internship or building on top of a larger project within your research group. If you're in a situation like this, and you share similar goals and values with the company or organization, then tech transfer is an effective way to get a version of your technology to a lot of people.

If tech transfer isn't viable, another path to public access is to create a commercial product. This offers greater control and freedom, but is significantly more challenging than building on an existing platform. Engineering your own product involves the same development challenges as in tech transfer, except you lack the resources of an established company. The rate of success for tech startups is extremely low, and there's a lot of competition. There are also often intellectual property restrictions that prevent students from forming companies or selling technology before they've graduated. Yet despite these challenges, there are established pathways for small-scale software production and many examples of students creating commercially viable software. Weebly was started by three Penn State students; Wordpress was co-founded by a freshman from the University of Houston; and Dropbox got its start while its founders were enrolled at MIT. Perhaps most famously, both Google and Yahoo were founded by Ph.D. students at Stanford University. The list goes on.

Deploying software isn't easy, but deploying physical technology involves additional challenges. Producing large quantities of hardware requires access to supply chains, manufacturing and assembly, and mechanisms for physical distribution. Despite these challenges, hardware production has become more democratic in the last five years. It is now feasible for small groups of people to manufacture and distribute hardware in a manner similar to early software production. Companies like Seeed Studio, with their in-house engineering, enable creators to scale products up from prototype to production runs of 1,000 units, and can scaffold the transition to Chinese manufacturing and distribution channels. U.S.-based companies AdaFruit and Sparkfun manufacture and distribute their own PCBs and kits, but they also manufacture and sell products for others, paying the original creators a royalty in the process. For example, the Lilypad Arduino and the Makey Makey, which both started as academic projects, were later manufactured and sold as products through Sparkfun. There are also a variety of options for generating funding. Crowdfunding platforms like Kickstarter, Indiegogo, and Crowd Supply can provide backing for small production runs. Innovation competitions, like the MIT $100K and Stanford's BASES Challenge, provide seed funding, publicity, and mentorship for successful applicants. Universities, such as MIT, have begun to recognize the educational value of enabling students to engage in manufacturing and have even developed classes that allow groups of students to travel to Shenzhen, China and learn about the manufacturing pipeline.

In the past, only large companies could effectively mass-produce technology. Today, small-scale hardware production is still very difficult. However new platforms and new educational opportunities have made it possible for students to mass produce and sell their own hardware and software, thereby enabling a broader range of people to gain access to it.

A third option is to deploy your technology as an open-source project. Releasing your work under an open-source license can stimulate deeper public engagement with your project; anyone can contribute new functionality and features, assist with documentation, and track down bugs. Although open source originated in conjunction with software development, it's also possible to open source physical designs. Electronic schematics, part lists, and CAD files can all be incorporated into an open-source project, enabling other people to reproduce and remix your technology using their own fabrication channels.

Despite their collaborative structure, open-source projects require a significant amount of maintenance and management. Without effective documentation, standards, and design guidelines you're unlikely to receive quality feedback and contributions. In addition, without careful evaluation of contributions, your technology may evolve in unintended and undesirable ways—potentially becoming too specialized or overloaded with features. The labor involved in managing open-source projects is compounded by the challenge of generating funding. Even well-established, open-source tools, can be difficult to maintain and fund—the Processing programming environment is but one example. Yet there are a growing number of options for financing open-source technology. Mozilla and Google Summer of Code offer grants to open-source creators to sustain or further develop their projects, while recurring crowdfunding platforms, like Patreon, enable anyone to support open-source projects through monthly donations. In addition, open sourcing your work doesn't necessarily prevent you from selling it as a product. Arduino, LittleBits, and Ultimaker are all examples of successful companies built around open-source hardware projects.

Sustaining an open-source project over the long term can be a lot of work. However if you're passionate about supporting innovation, open source can be powerful. Observing how people connect your work to unexpected domains and applications can in turn generate ideas for new tools and systems.

As you begin new projects, it's important to give yourself the freedom to explore and experiment. It's challenging to predict the directions a project may take in the long term, and sometimes it's productive to pursue research that is not intended for deployment. As you test out your ideas, however, it's worth reflecting on what you might gain by putting your prototype out in the world. We can learn a lot by developing and evaluating prototypes in the lab. Yet, when we go one step further and put our work out in the wild, we can shape the ways in which technology improves our world.

—Jennifer Jacobs

begin

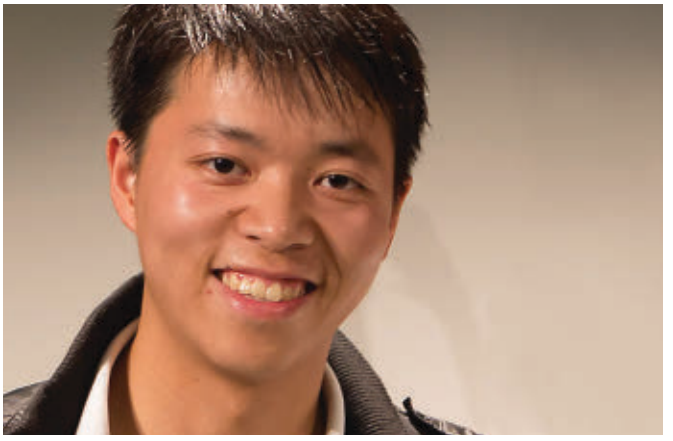
INIT

Quantum Computation: Double majoring in physics and computer science

The term "classical computing" might come as a surprise to many people. We have all heard of classical music, classic literature, and classicism: phrases that conjure up images of people in white wigs or archaic texts we were forced to read in high school. It might seem the term "classic" describes works or ideas that are antiquated, non-modern, or even obsolete, but this is not exactly what quantum computer scientists mean when they refer to contemporary computer science as "classical."

Quantum computing is a model of computation fundamentally different from the digital computers we are familiar with. The fundamental unit of information is not a bit but a qubit, which can take on a continuum of values between 0 and 1. It can be mathematically described by a point on the unit sphere in three dimensions where the two poles correspond to the 0 and 1 states. Instead of NOT and XOR logic gates, we have Pauli-X and CNOT gates, which are represented by unitary operators. We also have gates with no classical analogs, such as the $\pi/8$ and Hadamard gates. Clearly, we are dealing with a completely different computational model, and even a different paradigm of physics. It is this unique feature that attracts both physicists and computer scientists alike to this field. In this issue of *XRDS*, we take a closer look at this marriage of physics and computer science, which is profoundly influencing these traditionally disparate disciplines. Through articles written by experts in the field and reviews of recent advances, we will see how quantum computing is impacting areas such as computer simulation, complexity theory, simulated annealing, and even machine learning.

In addition to the crucial role it plays in computational physics, computer science has also contributed, in some unconventional ways, to physics through quantum computing problems.



At a fundamental level, it is clear physics has something to say about computer science since computers obey the laws of physics. Turing himself had a physical implementation of a computer in mind when he gave his mathematical definition of computation. This raises many intriguing questions: What are the physical limits on computation? Is there a gap between those limits and our current theoretical computational abilities? And, if so, what kind of physical system would saturate such limits?

In particular, what if our physical system is manifestly quantum? Can this help solve problems that are difficult for classical computers? Like many outstanding questions in computational complexity theory, such as the *P* vs. *NP* problem, this has yet to be answered with rigorous proof. But the evidence for "quantum supremacy" is quite substantial. A powerful example is Shor's algorithm; this polynomial time algorithm for integer factorization has substantial consequences for RSA cryptography. We also have the HHL algorithm.[1] Given a system of N linear equations, it can approximate $\vec{x}^{\mathrm{T}}M\vec{x}$, where M is a matrix and $\vec{x}$ is the solution to the linear equations, in time polylogarithmic in N. This is a common subroutine in more complex problems, and the algorithm is exponentially faster than the best classical algorithm, which takes time linear in N. Indeed, naïve classi-

1 Harrow, A. W., Hassidim, A., and Lloyd, S. Quantum algorithm for linear systems of equations. *Physical Review Letters 103*, 15 (2009), 150502.

A quantum computer could solve problems in a few minutes that would take a traditional computer roughly the lifetime of the universe.

cal algorithms would need linear time to even write $\vec{x}$ down. Applications of quantum computing even extend to online recommendation systems[2] and machine learning in general; a subject Bingjie Wang tackles in "Quantum Algorithms for Machine Learning." Another example is simulating quantum systems, such as molecules and complex materials, an idea Arghavan Safavi, Michael Wallace, and Martin Gärttner elaborate on in "Many-body Quantum Mechanics: Too big to fail?" Such algorithms are essential for drug design and the chemical industry.

The existence of these superior algorithms and protocols raises a deeper question: Is the complexity class defined by quantum computers a better candidate for the complexity-theoretic Church-Turing thesis? Stephen Jordan explores this question in "Black Holes, Quantum Mechanics, and the Limits of Polynomial-time Computability." He emphasizes that unlike analog computers, quantum computers are legitimate challengers since they do not require operations with exponentially high precision. Alexandru Paler, Austin G. Fowler, and Robert Wille explain how this is possible through quantum error-correcting codes. In "Reliable Quantum Circuits Have Defects," they focus on a particular class of codes that are robust against errors thanks to their topological properties. "Establishing Quantum Advantage" by Adam Bouland expounds on the general consequences of quantum computing for computational complexity theory, in particular applications to sampling problems.

Conversely, computer science also has much to say about physics. Like any other scientific discipline, physics has gained much from modern digital computing. However, in addition to the crucial role it plays in computational physics, computer science has also contributed, in some unconventional ways, to physics through quantum computing problems. Of these is the use of crowdsourced games to find optimized quantum algorithms that can run on realistic devices. In "Programming Quantum Computers Using 3-D Puzzles, Coffee Cups, and Doughnuts," Simon Devitt explains how this idea is realized in the puzzle game meQuanics.

Quantum computing is impacting areas such as computer simulation, complexity theory, simulated annealing, and even machine learning.

The tools computer science provides for physics go deeper than you might expect. Step into your school's center for theoretical physics today and you will hear these familiar words: "computational complexity." Modern developments in string theory and high-energy physics have uncovered connections between physics and computer science even at the highly theoretical level. Concepts like circuit complexity, error correction, and even unstructured database search have, via quantum computing, entered into the dialogue on black holes, wormholes, and quantum gravity in ways that no one could have ever foreseen. In "Black Holes and the Limits of Quantum Information Processing," Brian Swingle discusses a conjecture that connects a spacetime region's action, a fundamental quantity in physics that determines the equations of motion, to the circuit complexity of the problem of constructing a corresponding quantum state. Jordan also contemplates results that suggest a tantalizing connection between solving *NP*-hard problems in polynomial time by physical means and causality, the physical principle that prohibits superluminal signaling. Wrapping up this issue is Johannes Bausch's "Undecidability of the Spectral Gap." He explains how the existence of a spectral gap, a natural question in condensed matter physics and quantum annealing, is an undecidable problem.

From cryptography, error correction, and recommendation systems to black holes and superluminal communication, quantum computing provides a rich platform for discourse between computer scientists and physicists. The broader field of quantum information also brings together mathematicians, electrical engineers, material scientists, and even philosophers. And the technology inspired by these ideas attracts tech firms such as IBM, Microsoft, and Google, in addition to government agencies including NASA. Startups racing to invent the first practical quantum computer are appearing as well. The future outlook on quantum computing is indeed promising, but there is still much progress to be made. After all, the prestigious title of "neoclassical computing" is still waiting to be conferred.

—Dawei Ding,
Issue Editor

2 Kerenedis, I., and Prakash, A. Quantum recommendation systems. arXiv preprint. arXiv:1603.08675 (2016).

0.01°K A quantum computer machine needs to run in extreme conditions, at a temperature that makes it the coldest environment in our universe.

ADVICE

Time Management as a Ph.D.

As a Ph.D., you are effectively a student. But, unlike a normal student who takes classes, completes projects, and receives a grade, much more is expected from you. In addition to classes, Ph.D. students are expected to complete research, write papers, attend conferences, review papers, and even teach their own classes or assist professors in teaching their classes. And that is all before even considering the basic needs of daily life.

Ph.D.s quickly learn there is never enough time for everything they need to do. Because of this, time management is one of the most crucial aspects of successfully completing a Ph.D. As a Ph.D. student you must carefully consider what projects need to be completed first, and what can be pushed back. Here are some time management strategies you will need to master to be successful, along with a discussion on how to avoid common pitfalls.

Image by Katykin/Shutterstock.com

RULE #1: TIME IS YOUR MOST VALUABLE ASSET

The number one rule for Ph.D. students is guard your time jealously. You will meet many professors and other doctoral students working on fascinating projects. When you first begin your Ph.D., these opportunities will be everywhere. During your first year, the best thing you can do is be informed about who is doing what, but avoid taking on too many projects. Learn to say no to side projects that may potentially sidetrack you, and learn how to focus on those (two max) that will actually help you achieve your goals as a Ph.D. There will always be more projects.

Still, even if you just focus on your core project, you will inevitably be overwhelmed sooner or later. So let's discuss how to handle your day-to-day tasks and stay on track.

In my next article, we'll discuss how to know when it's time to quit your Ph.D.

RULE #2: KEEP RECORDS OF EVERYTHING

Pick out a calendar application, keep it updated, and check it every single day. My calendar is on Outlook, but Google Calendar or any other will do. In addition to keeping it updated, keep it synced to your mobile device, if you have one. Your calendar should have class times and meeting times included, but also consider blocking out study time and writing time. I also include paper deadlines, and usually project deadlines as well. This gives me a good overview of short-term and long-term goals. You can also color code blocks of time, to make your calendar visually digestible.

However, just starting a calendar isn't enough. You must check it each day, and stick to it. As soon as I have an appointment or meeting confirmed, it goes in my calendar. If it isn't in my calendar, it isn't happening. Without this mindset, you're just wasting your time creating a calendar.

$10-15 M The cost of a D-Wave Systems quantum computer.

RULE #3: MAKE CONCRETE PLANS, AND FOLLOW THEM

Make sure you use your calendar to plan out your day. When I arrive at my desk, the first thing I do is check my calendar. I know right away how much time I have at my desk, and how much time I have in classes or meetings. I can also begin to plan what project work I can accomplish during that time. Working some evenings and weekends is inevitable as a Ph.D., but I generally have an idea of what type of work I can complete at my desk, what I can complete at home, and what I'll have to spend extra time completing on the weekends. I usually prioritize coursework and teaching preparation, and once completed I work on research and papers. Coursework is usually easier than the research and papers; for me knocking out the easiest things first cuts down on my workload, boosts my morale, and makes it easier for me to concentrate on the more challenging research tasks.

But be careful. Many Ph.D. students fall into the trap of spending too much time on coursework and teaching, and let their research fall to the side. Be sure you make time each week for research and papers. You can't finish your Ph.D. without completing your research. If you find yourself losing interest in or even avoiding your research, it's time to have a talk with your advisor to discuss whether you need to switch directions, or even leave your program. (In my next article, we'll discuss how to know when it's time to quit your Ph.D.)

While planning out your projects, it's important to know how long it takes to do things. This is harder than it sounds, and is different for everyone, but it's an important skill to develop. It doesn't do any good to know that your paper is due in a week if you don't realize that it will take you two weeks to complete it. Knowing the amount of times things takes will give you the ability to plan ahead and begin working on things early. I plan out my time by looking at the project due date, and then working backwards. That requires knowing what steps need to be completed to finish the project, and approximately how long each will take.

FINAL WORD: DOWN-TIME!

Finally, prioritize sleep. There's an idea that being a grad student means that you have to give up sleeping. I find this idea really damaging. I can tell you from experience that if you make getting a good night's sleep your first goal each day, then you'll always be well rested. Your work improves, your mood improves, and your eating habits improve, improving your quality of life. Losing sleep some nights is inevitable, especially at the end of the semester or near important conference or journal deadlines, but losing sleep regularly to get your work done doesn't need to be a part of the Ph.D. lifestyle.

One last caveat: While a consistent sleep schedule is a major factor in maintaining your personal health, excessive sleeping can also be an issue. One of the first signs of depression can be sleep irregularities. Depression and other mental health issues are a consistent problem among graduate students, and maintaining your mental health is just as important as maintaining your physical health. If you find yourself having trouble sleeping, or other mental health symptoms, check your university's resources for on campus counseling. Many campuses offer free or reduced price counseling for students, and your personal health is far more important than any degree.

A Ph.D. is a marathon, not a sprint. Plan accordingly, and stick to the plan.

— Andrew J. Hunsucker

Blind quantum computing is a technique that involves mathematical traps to verify the outcome of a quantum computer when we have no other confirmation mechanism.

CAREERS

One Thousand Interviews

How customer insights keep one company agile, and challenge these data scientist to stay ahead in an ever-changing world.

Innovate or die—that is almost certain in today's business environment. Even the biggest names in industry have had to become "responsive to change," then "agile," and now they find themselves in a constant quest to create new value for their customers. Old business models may become worthless in the blink of an eye, getting big names into serious trouble or even a financial deficit.

We work for CGI, an IT and business services provider based in the Netherlands. CGI constantly redefines the value that we create for our clients. This is one of the reasons behind the annual review of our strategic plan, which involves employees, clients, and other stakeholders. The client side of this review requires almost 1,000 in-person interviews, where we listen to our clients perspectives, refine our thinking, inform our investments, and evolve our strategy to become an IT service provider of choice.

These interviews have clearly shown us that, globally, our clients are focused on becoming customer-centric digital organizations. Their top priorities include connecting with all stakeholders in order to become digital enterprises that can deliver the benefits of big data and business insights. However, they must always comply with continuously changing government and industry regulations. Furthermore, our clients often struggle to arrive at an outward-looking business case that is flexible enough to allow them to continue funding innovation. The costs of simply running their operations

Image by Igor Kisselev

The NSA's current cryptography requires hundreds of millions of qubits to crack it—a number far beyond near future projections for quantum computing.

tend to increase over time, detracting from opportunities to innovate and sustain themselves in the long run.

FINDING DIAMONDS

In order for us to help our clients grow, it is important for an IT and business services provider like CGI to offer a portfolio of digital solutions that reduce our clients' costs, while enabling digital transformation. Big data analytics is a key element of this portfolio, not by itself, but as a part of the applications that we offer.

To do this, we follow a four-step approach to designing and implementing a big data analytics solution, which is part of "Data2Diamonds." This is a CGI trademarked methodology covering the broad area of data and analytics. The approach requires a multi-skilled team that includes client experts, business analysts, developers, and data scientists. The team's goals are to inspire clients, identify and analyze high potential applications, and develop proof of value with live client data.

"Diamonds" are not easy to find. It takes involvement, knowledge, creativity, hard work, and passion to get the best out of your data. To that end, we use big data analytics to improve our clients' data quality and create predictive/prescriptive models that support their business models. It takes a passion for data, and we bring that to our clients, along with a commitment to help them succeed.

Our clients often struggle to arrive at an outward-looking business case that is flexible enough to allow them to continue funding innovation.

WORDS OF ADVICE

Computer science students are among a fortunate minority who can make a living doing work they enjoy, without worrying about being replaced by automation in the near future. Nonetheless, there are a few things to keep in mind before you start on your journey: Choose the right company, choose the right boss, don't undervalue mentorship, and don't be afraid to explore.

You will spend a large part of your life at work, so how you feel there will ultimately be very important. Instead of trying to find a job as soon as possible, take the time to explore all the potential companies within your network. For starters, the possibility of professional growth will always trump compensation, location, lifestyle, and so on. Some key indicators of growth at a company are the presence of a "young professionals program," grassroots projects, and innovative ideas.

But you will experience the most professional and personal growth if you have a great person mentoring you. Look for someone who is 10 times better than you in what you do (but humble and nice at the same time), and who gets satisfaction from sharing their knowledge and skills.

Remember the best mentor cannot hold your hand to guide you. You will have to put in the effort to create channels of growth. If you want to find diamonds, you better be ready to dig deep.

Biographies

Geerten Peek studied statistics and operational research in the Netherlands. He lives in the Nijmegen region. He is married and the father of two daughters. His current work includes CRM, BI, and big data analytics lead for manufacturing, retail, and consumer services at CGI.

Ahmet Taspinar studied applied physics at Delft University of Technology and is now employed as a data scientist at CGI. He blogs about machine learning fundamentals (www.ataspinar.com), and is the founder of Data Science Guild (ds-guild.nl).

The *XRDS* blog highlights a range of topics from conference coverage, to security and privacy, to CS theory. Selected blog posts, edited for print, are featured in every issue. Please visit xrds.acm.org/blog to read each post in its entirety. If you are interested in joining as a student blogger, please contact us.

BLOGS

CHI 2016 co-chairs Jofish Kaye and Allison Druin talk with keynote speaker Salman Khan, founder of Khan Academy.

Photo courtesy of Panagiotis Apostolellis

CHI 2016: Global, Diverse, Good

What can 1,000 scientists achieve when they invest one hour doing voluntary work?

By Nur Al-huda Hamdan

In the heart of Silicon Valley, the CHI 2016 conference broke through new ceilings. CHI (pronounced kai) is the most prestigious international conference in the field of human-computer interaction (HCI). It attracts researchers, designers, engineers, and artists who want to (re)shape technology and media to enhance people's quality of life. This year, the conference took place in San Jose, CA. More than 3,800 participants from 52 countries presented their work in various media formats: keynote presentations, media installations, interactive demos, and posters.

For her opening keynote, Nigerian-American journalist and author Dayo Olopade, portrayed the challenges she faced moving from the U.S. to Nairobi. She addressed how the demographics and culture of different countries are unique and should be taken into account in the design of new digital tools. Olopade took the audience on a voyage to Africa where she slowly dissolved the western lens, allowing attendees to see beyond chaos and desperation to reveal Africa's unconventional systems as an efficient act of "kanju"—a term that refers to the creativity that comes out of African difficulties. She showed areas in Africa where the informal infrastructure, streets and neighborhoods, did not make it into any map app or address book system. Her apartment in Nairobi "was best triangulated by using a Chinese restaurant, a petrol station, and an enormous pothole." Olopade encouraged the CHI community to view Africa in a more positive light . Instead of trying to westernize it with new tools, attempt to understand the continent as a whole and design *for* Africa. Although Amazon, Uber, and better postal services are needed in Africa, implementation cannot happen in the same way it does in Silicon Valley.

CHI touched upon other serious and global issues, such as the Syrian refugee crisis. Reem Talhouk and colleagues shared their own experience and research with Syrian refugees, and the challenges refugees face: including access to services, integration into host communities, and fleeing to safety. The panel also discussed how the research community can have a more actionable role toward aiding this emerging population, and emphasized collaborative research. In a time of increasing political and economical crises, Vasillis Vlachokyriakos and colleagues examined how HCI can promote democratic practices and social justice. They debated digital tools that open new avenues to alternative modes of political organization, civic

BLOGS

participation, and heightened awareness of the various power relations at play.

Education was one of the hottest topics at CHI 2016. Two other keynote speakers, Salman Khan of Khan Academy and Kimberly Bryant of Black Girls Code, spoke of new ways to deliver quality education to everyone. Khan pointed out online education should complement the physical classroom and not replace it. In turn, Khan Academy is working with several schools to test new teaching techniques and materials that would enhance students' learning experience by combining the two learning methods.

In the words of the conference chairs, CHI 2016 was "a more humane conference, transparent, data-driven, and accounted for the importance of families and work/life balance." Women had a strong presence in all disciplines and positions. They were (vice)presidents, professionals, professors, doctoral candidates, and students in attendance. Making CHI one of the most diverse scientific conferences out there. For the last three years, the conference hosted the CHI Women's Breakfast for about 100 attendees to celebrate women in computing and discuss the gender gap in computer science fields. This year the conference renamed this event; the "Diversity and Inclusion Lunch" had 500 diverse attendees. Aspects of diversity were expanded beyond gender to include aging, disability, physical appearance, race, ethnicity, nationality, marital status, and mental health. Speakers shared their personal stories and explained how they stood up in the face of these challenges by establishing support groups and speaking up. But inclusion may not be as simple as diversity reports on company dashboards. Karen Holtzblatt moderated a panel on the status and challenges of minorities in high tech. While many companies have implemented new recruitment techniques to reduce bias against underrepresented groups, Holtzblatt called on companies to take the next step and understand the experiences of these groups so they would "stay, advance, and thrive." At the conference level, the organizers did just that by providing parent attendees with free-of-charge child care at the conference site.

A core framework in HCI is "user-centered design"—understanding your users, design, evaluate, analyze, and iterate. Each year, CHI organizers take this concept for a test drive holding several sessions, such as the ACM SIGCHI Town Hall Meeting and CHI Chairs Ask Me Anything (AMA), to discuss with the community how the conference programs, the review process, as well as the publication and dissemination channels could be enhanced. This year, in "Transparent Statistics in HCI," session attendees brainstormed new ways to inquiry the quality of data acquisition and analysis in research papers, and encourage authors to publish their data and replicate other studies.

The CHI 2016 theme was "chi4good." For the first time, conference organizers cracked the shell that separates scientific conferences from surrounding communities. Attendees were asked to arrive one day before the beginning of the conference to spend a few hours partaking in community work. As a result, more than 700 hours were spent volunteering for local nonprofits in the Silicon Valley area.

CHI 2016 demonstrated how the impact of the science community can traverse beyond paper format to affect people in their current environments. Conferences of different fields should work on engaging scientists in more community and volunteer work to bridge the gap between incremental science and people's real and current needs. Encouraging different institutions to work together to do good leads to new networks, an exchange of expertise and knowledge, and future collaborations.

To this day, the field of HCI and the CHI community have put forward proposals to address many real-world problems. These solutions still need more investment and must reach the right people for them to have the desired impact. But what about prevention tools? How can HCI equip people with tools that help them avoid rather than merely cope with crises? How can HCI help people become more informed of their economical and political spheres, or even the "fine print" on the products they purchase on daily basis? These are questions the CHI community will have to answer

After five days, CHI 2016 concluded. Attendees left the San Jose Convention Center with new aspirations and inspirations. CHI has been celebrating research in the field of HCI for the past 34 years. But this year, the conference set out new goals with a global scope. The evidence presented this year points out the intricacies of different populations, and how targeted design is more impactful than one-design-fits-all.

To access the conference keynotes and presentations: check out the YouTube channel "acmsigchi." And for access to the conference proceedings, check out the ACM SIGCHI conferences website; http://www.sigchi.org/publications/toc/.

Biography

Nur Al-huda Hamdan is a Ph.D. candidate and research assistant at RWTH Aachen University, Germany. She has a background in computer science and engineering. She does HCI research on wearables and interactive textiles.

143 km The teleportation of entanglement travelling this distance proves the feasibility of a quantum repeater in a space- and ground-based worldwide quantum Internet.

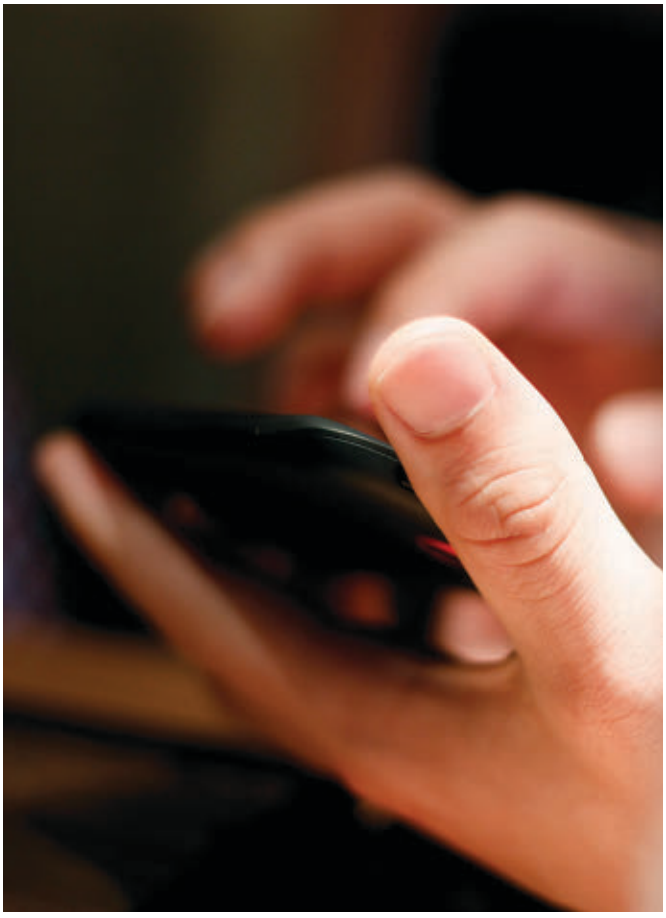


Image by Graeme Paterson

An Introduction to Gamification in Human-Computer Interaction

Improving user experience through game play.

By Gustavo Fortes Tondello

User experience (UX) is a field within human-computer interaction (HCI) that studies the whole experience of a user with a product, system, or service. UX focuses on issues such as usability, ergonomics, cognitive load, and affective experiences. However, in the last few years, there has been a growing interest in understanding users' motivation to use a product, system, or service. This interest is spawned by observable low engagement rates: It is not enough to have a useful system, one needs to also motivate and engage users in it. One possible solution to this comes from a field of study called gamification or gameful design.[1] Its main inspiration comes from understanding the factors that make games fun and motivate people to play them voluntarily with so much engagement.

Gamification is defined in HCI as "the use of game design elements in non-game contexts" [1]. There are two important concepts embedded in this definition:

- *Game design elements:* The parts used to build games. In this context, we refer to the parts that afford the gameful experience, instead of the technologies involved in creating the game. Thus, we are not interested in things like graphics and audio. Instead, gamification focuses on elements such as challenges, levels, avatars, points, achievements, stories, and leaderboards.
- *Non-game contexts:* Those applications whose main purpose goes beyond pure entertainment. Examples of contexts where gamification has been applied include: business, marketing, education, and health.

Deterding et al.'s definition [1] also suggests gamification consists of using game elements in a system that is not a full game. This is different from serious games, which are also used in non-game contexts but with a different approach. Gameful design also differs from playful design because the former focuses on activities that are oriented to goals and structured by rules, while the latter focuses on free-form and improvisational activities (although both gameful and playful design can be applied together to the same product). Figure 1 situates gameful design between the poles of games and play, parts and whole.

Most gamification researchers have been seeking to understand users' motivations to interact with a product or system by means of the self-determination theory (SDT) [2]. SDT posits human beings can be intrinsically or extrinsically motivated to engage with any task. Intrinsic motivation refers to wanting to do something just because the task itself is enjoyable. Extrinsic motivation refers to doing something because there is a possibility of achievement, some additional outcome, such as earning a reward or fulfilling an obligation. Furthermore, SDT posits intrinsic

1 There have been a few different definitions of gamification and gameful design from different fields and authors. We have also seen some heated discussions attributing slight different meanings to these terms and arguing in favor of one or the other. However, we use both terms here from the point of view of HCI research and attribute them both the same meaning.

BLOGS

motivation is supported by activities that fulfill three psychological needs: competence (feeling capable of doing something), autonomy (feeling free to choose how to do something), and relatedness (feeling connected with other people). SDT researchers have demonstrated the fulfillment of these three psychological needs can explain why players enjoy games so much [2]. For example, completing quests or beating a difficult boss in a game makes the player feel competent. Being able to choose different paths or to create different things makes the player feel autonomous. Finally, playing with other people (in cooperation or competition) makes the player feel related. Thus, these insights have often been applied to gamification by selecting and using game design elements that can lead users to feel the same kind of motivation when interacting with any system.

Figure 1. Gamification between games and play, parts and whole.

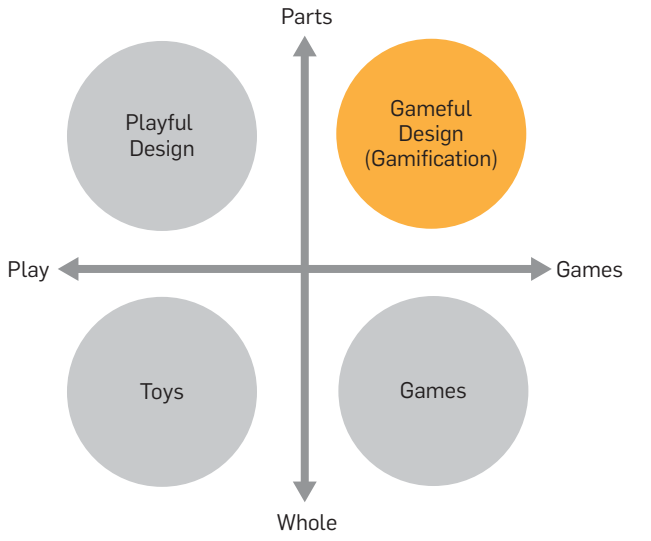
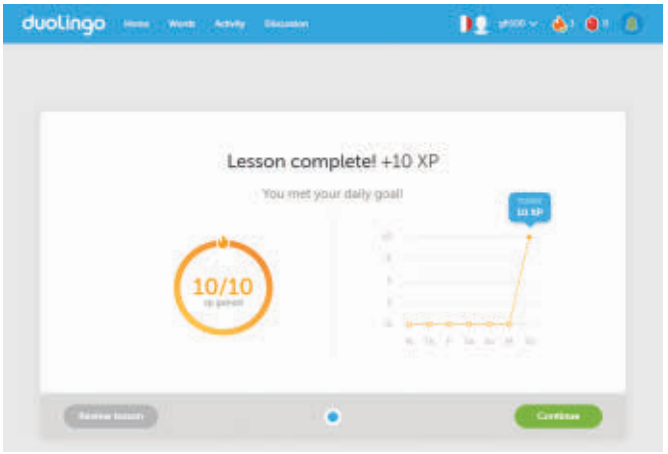


Figure 2. Gameful elements after my first completed lesson at Duolingo.



An example is the language-learning site Duolingo. Figure 2 shows how Duolingo used gameful design elements after I completed my first French lesson. Before I began, the application allowed me to choose the language I wanted to learn, how much time I wanted to study per day, and if I wanted to begin at the basic or the advanced level. These choices helped me feel autonomous. While doing the first lesson, a progress bar was always visible showing me I was getting closer to achieving my goal. After I completed the lesson, I was informed I had completed my daily goal and earned experience points. All of this helped me feel competent. The daily streak counter (the fire icon at the top right) also motivated me to engage with the application every day. Finally, it is possible to connect with other users inside the platform, helping me feel related with others. Duolingo has been cited as an interesting example of gamification, and its learning effectiveness has been independently studied (https://www.duolingo.com/research).

In HCI, the study of gamification has often been part of the sub-domains of player-computer interaction (PCI) and player experience (PX), which study the experience of players interacting with games. Research focused on games with a purpose (serious games) and gamification has been increasingly popular at the ACM CHI conference, as well as the recently created ACM CHI PLAY Conference, which is focused on the PCI sub-domain. Furthermore, Gamification 2013 was a focused conference held at the University of Waterloo that put together scholars interested in gameful design research and applications.

Despite its popularity, gamification research is still an emergent field and much remains to be done. A review by Seaborn and Fels in 2015 [3] noted usage of the term gamification remains inconsistent; more empirical, mixed-method research that reports statistical analysis and effect sizes are needed to substantiate the initial positive effects reported. Furthermore comparative studies with controls are needed to ascertain what effects gamification has beyond other approaches. Another review by Hamari et al. in 2014 [4] suggested gamification does work, but some caveats exist as most quantitative studies reported only partially positive results. The reasons for this still need to be further investigated.

A quantum computer could easily crack a security code that would otherwise take thousands of years using the most powerful supercomputers.

Besides additional investigation regarding the results of gameful design implementations, more research is also needed regarding gameful design methods. Many design methods have been described by industry practitioners, but these often lack a solid theoretical foundation and proven empirical results. Seeking to fill this gap, Deterding has reviewed several industrial and academic gameful design methods and proposed the "lens of intrinsic skill atoms" [5]. This is a design method backed by scientific research on motivation and game design, and has been applied in several case studies. Deterding's method focuses on identifying the underlying challenges of the activity and helping the user reframe them as gameful challenges, with help of motivational design lenses. Nicholson introduced the term "meaningful gamification" [6], which aims to help a user find personal connections that motivate engagement with a specific context for long-term change. This is achieved by employing six new concepts in gameful design instead of a reward-based design: reflection, exposition, choice, information, play, and engagement. Kappen and Nacke introduced the "kaleidoscope of effective gamification" (KEG), which describes several design layers that need to be applied to a gameful system to achieve effectiveness—in this context this is described as "the successful engagement of a player through effective game design" [7]. KEG describes four layers: the motivated behavior layer, the game experience layer, the game design process layer, and the perceived layer of fun.

Finally, another topic that has been recently receiving attention is the personalization of gameful applications. Several studies have suggested different people respond differently to gameful applications; thus, a personalized approach seems to be more engaging than a one-size-fits-all approach. This topic was investigated in the Workshop on Personalization in Serious and Persuasive Games and Gamified Interactions" (http://personalizedgames.tech-experience.at/) held during ACM CHI PLAY 2015. One approach for personalization in gameful design is understanding and tailoring the design to a particular user's motivations and personality. Among the diversity of player and user type models in the literature, there are two recent research-based models that can be used for this approach. One of them is the BrainHex model [8], which is based on neurobiological research and describes seven types of players according to motivation: achievers, conquerors, daredevils, masterminds, seekers, socialisers, and survivors. Another is the gamification user types Hexad [9], which is based on the theories of intrinsic and extrinsic motivation and describes six types of users in gameful systems: achievers, free spirits, philanthropists, socialisers, players, and disruptors.

Gamification is an interesting and exciting research topic in HCI. Initial results have shown it carries great potential for improving engagement in user experience and positively helping people and businesses achieve their goals. Nevertheless, there are still open research questions to be explored. Uncountable practical applications are being implemented all the time and reporting favorable results, despite often lacking scientific validation. Because of all these factors, we expect to see many valuable results from gamification research for the following years. Stay tuned!

Completing quests or beating a difficult boss in a game makes the player feel competent. Being able to choose different paths or to create different things makes the player feel autonomous.

References

[1] Deterding, S., Dixon, D., Khaled, R., and Nacke, L. From game design elements to gamefulness: defining "gamification". In *Proc. MindTrek '2011* (pp. 9-15). ACM, New York, 2011.

[2] Rigby, S., and Ryan, R. M. *Glued to Games: How Video Games Draw Us In and Hold Us Spellbound*. ABC-CLIO, 2011.

[3] Seaborn, K., and Fels, D. I. Gamification in theory and action: A survey. *International Journal of Human-Computer Studies 74* (2015), 14-31.

[4] Hamari, J., Koivisto, J., and Sarsa, H. Does gamification work? A literature review of empirical studies on gamification. In *Proc. HICSS '2014 IEEE*. Jan. 2014, 3025-3034.

[5] Deterding, S. The lens of intrinsic skill atoms: A method for gameful design. *Human-Computer Interaction 30*, 3-4 (2015), 294-335.

[6] Nicholson, S. (2015). A RECIPE for meaningful gamification. In *Gamification in Education and Business*. Springer International Publishing, 2015, 1-20.

[7] Kappen, D. L., and Nacke, L. E. The kaleidoscope of effective gamification: deconstructing gamification in business applications. In *Proc. Gamification '2013*. ACM, New York, 2013.

[8] Nacke, L. E., Bateman, C., and Mandryk, R. L. BrainHex: a neurobiological gamer typology survey. *Entertainment Computing 5*, 1 (2014), 55-62.

[9] Marczewski, A. User Types. In *Even Ninja Monkeys Like to Play: Gamification, Game Thinking and Motivational Design* (1st ed.). CreateSpace Independent Publishing, 2015, 65-80.

Biography

Gustavo Fortes Tondello is a Ph.D. student in computer science at the University of Waterloo, Canada.His main interests include gamification and games for health and learning. His research focuses on the design of gameful applications.

UPDATES

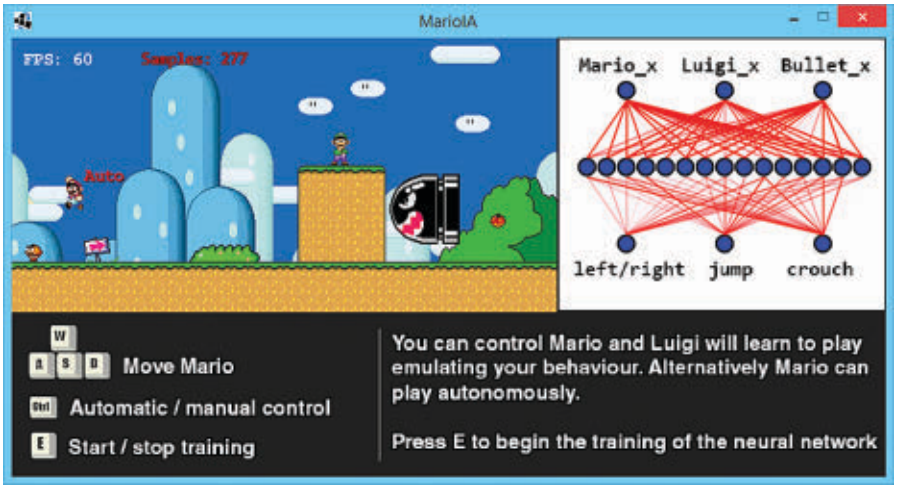
"ANN" Helps Mario Rescue Princess Toadstool

In the middle of the 2013–2014 academic year, some colleagues and I started a new ACM chapter at the University of Salamanca in Spain. We launched the association at the beginning of exams season; but unexpectedly we survived and made it through with our organization intact.

In April 2014, we sponsored our first chapter activity: introductory workshops on Arduino, Perl, and other interesting technologies. But what I did not know was at the time I would discover what would become one of my favorite knowledge areas within computer science: artificial intelligence, more specifically, artificial neural networks (ANNs).

Simply put, an ANN is a set of artificial neurons (information processing units) arranged in layers and interconnected to each other so they can process information and then forward it to the next layer of neurons. In short, ANNs enable machines to learn.

The most popular kind of ANN is the "Feedforward" ANN, inside of which information travels only in one direction, so the network has a clearly defined input and output. If the weights of connections between neurons are adjusted to the right value, the ANN can approximate any function. That is why we can consider ANNs as universal function approximators. But, of course, we have to find those optimal weights first, and that can be a difficult task.



This is where the world of Mario, Luigi, and their beleaguered princess come in.

In September 2014, our chapter decided to introduce a workshop related to ANN techniques. We wanted to show an interesting, visually attractive, real-world application. After some research on typical ANN application fields, we came up with a great idea: Showing how ANNs can be applied to a field we all love, video games. We chose Super Mario and tried to write a simplified version of it. After a couple of weeks of Java programming, we had a bounded part of the Super Mario world, populated by Mario, Luigi, a Goomba, and one of those hateful giant bullets (to make things more tricky). The goal for Mario was to smash the Goomba while simultaneously avoiding Bullet Bill We created a unified player interface that could be extended to implement different kinds of player agents; the idea was to have a human player controlling Mario for a while, which would allow us to get information to train the ANN.

We decided to use the simplest type of Feedforward Neural Network, the multi layer perceptron (MLP). It had three layers in total. First, an input layer with three neurons representing the position of Mario, the Goomba, and Bullet Bill (of course these values were normalized first). Then we had what is called the "hidden layer," formed by 15 neurons, and finally the output layer with again three neurons representing the actions Luigi could perform (i.e. move left or right, jump, or crouch). When an input vector of data was presented to the network, it computed the output and Luigi was commanded to perform the action corresponding to the output neu-

ACM USAL at a Glance

School: University of Salamanca (Spain)

Chapter Name: ACM USAL

Location: Spain (various cities)

Website: http://usal.acm.org/;

Facebook: https://www.facebook.com/ACMUSAL/

Date Established: Jan 2014

Officers: Juan Alberto García Esteban, President; Héctor Gonzalo Andrés, Vice-president; Emilio Cobos Álvarez, Secretary

Current Total Membership: 50

Contact: acm.usal.chapter@gmail.com

The D-Wave 2X Systems allows for a search of 21,000 possibilities—which is higher than the total number of particles in the universe—thanks to a lattice of 1,000 qubits.

ron with the strongest activation level (the highest output value).

In the beginning, Mario had to be controlled by a human player, and the system would store vector pairs corresponding to the neural network input (character positions) and the desired output (human-like behavior). When enough data was stored, the training procedure could start and by the end of this process the weights of the neural network were fixed, the network would have learned the right internal configuration needed to emulate a human player. At this point Mario would leave the game and Luigi would automatically start to play based on the recently learned behaviors from the collected data. We added some graphical representations of this internal weight configuration and its evolution over time, finally we had a workshop ready to go.

The workshop took place on October 16, 2014. William Raveane, a colleague of mine working on his Ph.D. with ANNs, presented neural networks from a theoretical perspective. I developed the practical part of the workshop. At the end, we were pleased with the results, and I think our audience was, too. Since then, we have held workshops on ANNs each academic year, with more attendees at each session.

If you are interested, the code used for the workshop and the executable .jar files can be found here: https://github.com/lopeLH/SuperMarioMLP.

Biography

Daniel López Sánchez is a predoctoral student at the University of Salamanca. He holds a bachelor's degree in computer engineering and a master's degree in intelligent systems. Most of his work centers around low computational cost machine learning, deep learning, and artificial vision. He is one of the founding members of the ACM USAL student chapter.

MILESTONES

Quantum Milestones

Early 1980s Researchers Paul Benioff, Yuri Manin, and Richard Feynman independently investigate computer models that are able to simulate quantum systems, which are the first conceptions of quantum computing.

1985 David Deutsch of Oxford University publishes a paper describing the first universal quantum computer, which uses "quantum gates" to behave similarly to a universal Turing machine.

1994 Peter Shor, working at AT&T, proposes an algorithm using qubit entanglement and superposition to find the prime factors of any integer. Shor's algorithm works in polynomial time, and is thus far more efficient than any other factoring algorithm at the time.

2001 Researchers at IBM Almaden and Stanford University are the first to successfully execute Shor's algorithm to factor the number 15, using a seven qubit computer.

2016 IBM Research announces the "IBM Quantum Experience" for the public to work hands-on with a as a cloud-based system composed of five superconducting qubit computer.

—Jay Patel

Quantum Algorithms for Machine Learning

Quantum computing and machine learning are two technologies that have generated unparalleled amounts of hype among the scientific community and popular press. Both are mysterious, immensely powerful, and on a collision course with each other.

By Bingjie Wang

DOI: 10.1145/2983535

Quantum computing is computing with the laws of the quantum mechanics. When physics is taken to a very small scale, the common intuitions from large-scale, or classical, mechanics fail. Experiments performed in this small-scale world show our large-scale equations are approximations erasing the possibility of something existing, both as a particle and a wave. Currently, quantum mechanics is a hindrance for classical computing. If transistors were any smaller, then quantum effects come into play. However recent research has turned the tables to ask "What can quantum mechanics do for computing?", instead of "What can computing do about quantum mechanics?"

In 1992, David Deutsch and Richard Jozsa proved quantum computing could do something that classical computing could not. Let's consider a function with 0 or 1 as an input that returns 0 or 1 as its output. The function is either balanced: returning both 0 and 1. Or it's constant: returning only 0 or 1. The possibilities are listed below:

```
  ▶ f(0) = 0, f(1) = 0.
Constant zero function.
  ▶ f(0) = 1, f(1) = 1.
Constant one function.
  ▶ f(0) = 0, f(1) = 1.
Identity function.
  ▶ f(0) = 1, f(1) = 0.
Negation function.
```

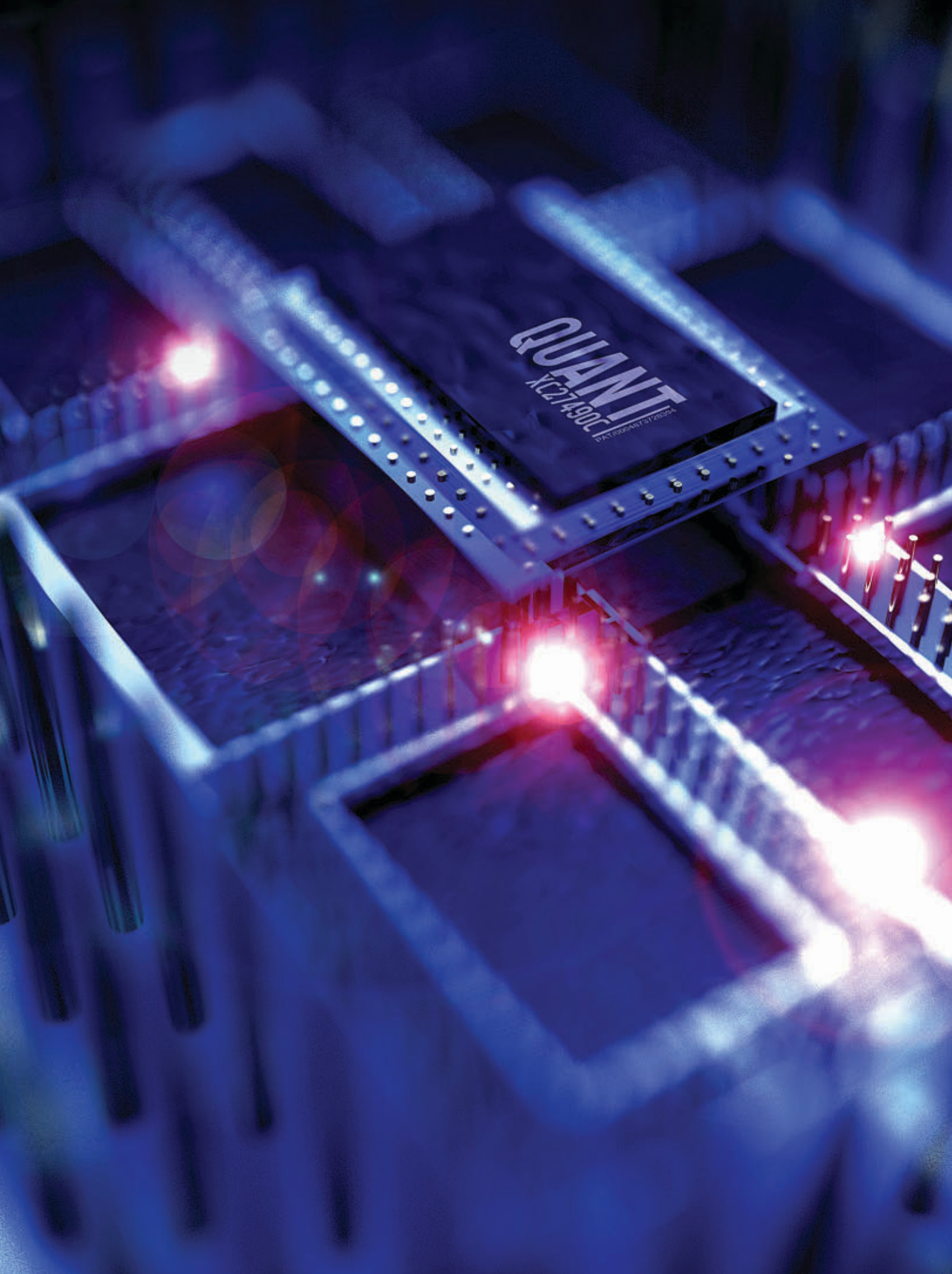
The question is: How many queries are necessary to determine whether it was balanced or constant? In classical computing, two queries are needed, but with quantum computing's Deutsch-Jozsa algorithm, a single query is sufficient.

Practically speaking, this isn't much, but it broke new ground in showing quantum mechanics had powers to be exploited. It did not take long for a game-changing application to arise. In 1994, Peter Shor devised a quantum computing algorithm that could efficiently factorize large numbers. Modern cryptography hinges on the fact that multiplication is easy, but finding factors is hard. With the potential to shatter cryptography, Shor's algorithm galvanized research interest in quantum computing.

Embracing quantum computing means embracing the probabilistic nature of quantum mechanics. Coincidentally, in 1992, the Fifth Generation Computer Systems project was abandoned. This was a gargantuan-scale artificial intelligence project based on logic programming. Roughly speaking, the belief behind logic programming is that artificial intelligence could be achieved through logical reasoning. While this approach has had notable successes, such as Garry Kasparov's defeat by the Deep Blue chess engine, the everyday world is rather unkind. One challenge is commonsense knowledge; as humans, we manage a huge deal of information that we don't notice. However this type of processing is just not practical for computers. In the face of this and other challenges, artificial intelligence turned toward probabilistic methods. The research would eventually evolve into the field of machine learning.

The historical parallels of quantum computing and machine learning are somewhat uncanny. Though, there is more than history and some handwave

Image by Welcomia/Shutterstock.com


QUANT
XC27490C

notions of "probabilistic" nature. Research is currently striking at the heart of an immensely practical question: Can quantum computing be used for machine learning?

COMPUTING WITH INTERFERENCE

What makes quantum mechanics different from classical mechanics? This is a difficult question. As Niels Bohr says: "We must be clear that when it comes to atoms, language can be used only as poetry." In lieu of heavily mathematical discussions, I choose to use the many-worlds interpretation, summarized by Vaidman [1]. But two disclaimers are required. First, as Bohr says, this type of discussion can only paint an incomplete picture and ultimately needs to be grounded in mathematics. Second, I am not an expert in such interpretations.

It should be noted there are other interpretations, such as the Copenhagen interpretation, which is better known as "shut up and calculate." Which interpretation is correct is subject to debate, but these are merely interpretations of what physicists observe in experiments. The practical implications for quantum computing are agnostic to the interpretations explaining them.

In the many-worlds interpretation, multiple realities exist at the same time and we are observing a particular reality. Let's follow Schrodinger's example and place a cat in a box with a poison designed to activate at random. With the many-worlds view, when the box is opened the observer could exist in a reality where the cat is dead or where the cat is alive. The idea that there is more than one reality is not specific to quantum mechanics. What is new, is the idea of "interference." This concept originates from waves. Consider two water waves meeting at the beach, if a crest meets another crest, the wave gets bigger. This is constructive interference. Likewise, destructive interference is when crest meets trough, cancelling each other out. With respect to the many-worlds interpretation, one crucial difference between quantum and classical mechanics is realities interfere like waves. Even two paradoxical realities, such as dead cat and alive cat realities, can interfere, therefore the cat is both dead and alive!

For computing, a quantum bit, or qubit for short, is either zero, one, or a superposition of "realities"—meaning it is partly zero and partly one with specified amplitudes. A higher magnitude of amplitude means that at a higher probability we observe a particular value. But unlike probability, amplitude can be negative, allowing for destructive interference. A quantum state is a collection of qubits.

Now the stage is set to describe the Deutsch-Jozsa algorithm. First, set up a qubit with equal superposition of zero and one, with the intention of querying the function with both zero and one to the function in different "realities." Next, a quantum "logic gate" is enacted, creating destructive interference if the function is balanced and constructive if the function is constant. Finally, the box is opened by connecting the signal to a wire. If there is no signal, then the function is balanced, otherwise the function is constant. As Niels Bohr reminds us, "words are only poetry." The interested reader should consult Nielsen and Chuang's *Quantum Computation and Quantum Information* (2010) for the Deutsch-Jozsa algorithm and other magical quantum tricks, such as quantum teleportation [2].

Harnessing the power of interference requires careful control of interference. There aren't any quantum "cats" running around, because large-scale objects like cats interact and interfere with many more objects than small-scale objects like electrons. The interference drowns out quantum mechanics. This problem is known as "decoherence," and is the main challenge for implementations of quantum computers.

With the potential to shatter cryptography, Shor's algorithm galvanized research interest in quantum computing.

MODELS DETERMINED BY DATA

While quantum mechanics is counterintuitive, paradoxical, and confusing, it has held the attention of top physicists for more than a century. As a result, mathematical tools are available to answer research questions about quantum mechanics and its applications to computing. Machine learning, on the other hand, is a much younger field. Machine learning approaches work, and spectacularly so, but little work has been done to understand why. In my view, Bishop's *Pattern Recognition and Machine Learning* (2006) presents a good balance between the mathematical basis for machine learning and the broad array of machine learning techniques.

Machine learning is typically divided into three types of tasks: supervised, unsupervised, and reinforcement. Of the three, only supervised learning has been explored alongside quantum computing. Though, it's likely only a matter of time before links are made there as well.

In supervised learning, the setup works as follows. There is a black box we would like to estimate using a possibly noisy dataset containing some inputs and their corresponding outputs. The input can be described using a vector of features. The dataset itself can be described by the feature matrix and target vector, formed by stacking the feature vectors and target value respectively. We hope the dataset is sufficiently representative, so the estimate we build from is general enough to be used for predicting target values on previously unseen inputs.

Let's consider self diagnosing an embarrassing diseases. In our case, the features are the symptoms and the target is whether you should seek professional advice for a particular medical issue. The dataset would consist of a sample of patients and their diagnosis by medical professionals, which, at times, can be faulty. Traditionally, statistics has been the tool for such applications: assume a specific model for the black box and infer the parameters based on the dataset. However, as

Bishop insists, it is necessary to adapt the model to the data. This is the crux of machine learning. For example, *k*-Nearest Neighbors looks through its dataset, finds at the *k* examples whose feature vectors are closest to the input feature vector, and outputs an average of the *k* example's target values.

Support vector machines can be viewed as a generalization of *k*-Nearest Neighbors. Instead of *k* neighbors contributing to the prediction, every example in the dataset contributes, but far-away examples contribute significantly less. Instead of a simple average, the output is the weighted sum of the contributions. These weights are chosen with a "curve fitting" algorithm to optimize the fit between the model's target values with the dataset's target values. The key idea is the optimization attempts to set as many weights as possible to zero, sometimes trading fit in order to do so. After the optimization, the feature vectors corresponding to non-zero weights are known as the "support vectors" upon which the model is based on.

Statistics has many weaknesses. For one, imposing a model on the black box creates bias. Then there's "data-dredging" and "p-hacking." Machine learning leaves these decisions to the optimization algorithm. However, the cost is that when asked what intuitively makes the support vectors special, machines can offer little better than: "the optimization algorithm said so." Theoretical results such as the Karush-Kuhn-Tucker condition help, but there's much more to be understood.

QUANTUM SUPPORT VECTOR MACHINES

The previous section introduced the *k*-Nearest Neighbors algorithm. The performance is dependent on the performance of finding what vectors are in the neighborhood. This is especially problematic when the dataset is very large.

In 1996, Lov Grover proposed his quantum search algorithm. Grover's algorithm considers a function that accepts arbitrary binary string as input and returns true or false. It can assume only one string returns true. If there are more, interpret binary strings as numbers and ask for the smallest string, which the function returns as one. Then, rerun the algorithm, except this time looking for the second smallest, and so forth.

Research has shown the potential of quantum computing for machine learning, but why hasn't this been implemented yet?

The algorithm borrows many ideas from the Deutsch-Jozsa algorithm. Again, the interested reader should consult *Quantum Computation and Quantum Information*. First, set up a quantum state with equal amplitudes for every possible input. Then the "realities" where the function returns zero are iteratively interfered out. In the worst case, a classical algorithm requires a query for every possible input. Grover's algorithm's performance is the number of interference iterations. In addition, there is a failure probability, so a query needs the output to be checked. If it's wrong, the algorithm needs to be re-run. All in all, in the average case, Grover's algorithm is quadratic improvement over its classical counterpart.

Grover's algorithm can directly be applied to improve the *k*-Nearest Neighbors to search for the feature vectors in the dataset that are closest to the input. This is impressive, but is there more? For the most part, *k*-Nearest Neighbors is only of historic interest.

In 2008, Aram Harrow, Avinatan Hassidim, and Seth Lloyd proposed the HHL algorithm [4]. The algorithm prepares a state whose amplitudes correspond to the values that solve a system of a system of equations. The intuition behind HHL is based upon Feynman's 1982 motivation for quantum computing: Quantum computers can simulate quantum mechanics much faster than classical computers. After some rewriting, a system of equations can be seen as a Hamiltonian: An operator describing how energy changes in a system, acting on a quantum state, which can then be simulated using a variant of Feynman's algorithm.

Now, if there are *n* equations in the system, the number of quantum "logic gates" HHL requires is proportional to the logarithm of *n*. Whereas classical computers can do little better than Gauss-Jordan elimination, which is proportional to *n*-cubed. This is an exponential improvement, blowing Grover's quadratic speedup out of the water. It's hard to overstate the magnitude of this achievement. Solving systems of equations is applicable to much more than machine learning. It's a fundamental problem that arises in almost all of science.

Then, in 2013, Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost discovered a way to perform principal component analysis [5]. Like HHL, quantum principal component analysis is also based on Feynman simulation and provides an exponential improvement over classical algorithms. This is the final ingredient for a quantum implementation of support vector machines. Rebentrost, Mahsoud, and Lloyd's quantum support vector machine algorithm [6], uses a subroutine from quantum principal component analysis to create a HHL-compatible version of the Gram matrix (matrix of pairwise contributions), then the curve fitting algorithm used to optimize the fit can be converted into a set of linear equations that is solved using HHL.

However, Andrew Childs points out the caveats, or fine print according Scott Aaronson, behind these algorithms [7]. HHL prepares a quantum state but doesn't put signals to a wire. Obtaining the support vectors takes linear time and the exponential improvement is lost. However, this doesn't preclude using the solution state as a subroutine for other quantum algorithms.

In addition, simply reading the system of equations costs *n*-squared steps. So the Gram matrix must be sparse. But even sparsity isn't enough, it also has to be "nice," in the sense that its eigenvalues must be approxi-

While quantum mechanics is counter-intuitive, paradoxical, and confusing, it has held the attention of top physicists for more than a century.

mately equal. Likewise, the target vector must also be sparse, or at least, relatively uniform. However it is not clear these conditions hold in real-world applications of support vector machine learning.

WHAT IS NEXT?

As machine learning moves to neural networks and deep learning, work is underway to see how quantum computing could improve these techniques. Unfortunately, the theoretical foundations of deep learning are poorly understood. It's not even clear what makes deep learning deep. There isn't a precise formulation of the classical problem for quantum computing to tackle.

An alternative is to turn around and ask what machine learning can do for quantum computing. However, machine learning is used when the black box is too difficult to determine, and for quantum mechanics the black box isn't so black. Although, as experiments become increasingly complex and intractable, machine learning is becoming an increasingly attractive option over scientific computing and traditional mathematics. This is the approach taken by Wiebe et al. for Hamiltonian estimation [8]. Another interesting approach is the one taken by Bisio et al. [9]. Their work uses a combination of classical computing and quantum computing to learn the action of an unknown quantum logic gate.

Almost never before has there been a technology as widely applicable as machine learning. In our daily lives, it's almost impossible to not come in contact with machine learning: Your location, purchases, and search history are constantly being data mined and learned. Research has shown the potential of quantum computing for machine learning, but why hasn't this been implemented yet?

Currently, the best quantum computers have, at most, a dozen high-quality qubits or a few thousand low-quality qubits. Machine learning, on the other end, is where terabyte datasets are not uncommon. As more applications for quantum mechanics are discovered, there will be more interest in fighting decoherence and implementing quantum computers. While there is a lot of work to do, the future looks bright. Recently, Canadian Prime Minister Justin Trudeau pledged money to the Perimeter Institute for Quantum Computing. When asked by a reporter to explain quantum computing, Mr. Trudeau gave a clean minute-long explanation. I don't think any technology, even machine learning, has had that kind of honor.

References

[1] Vaidman, L. Many-world interpretation of quantum mechanics. *The Stanford Encyclopedia of Philosophy*. Spring 2016 edition. 2016.

[2] Nielsen, M. A., and Chuang, I. L. *Quantum Computation and Quantum Information*. Cambridge University Press. 2010.

[3] Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] Harrow, A. W., Hassidim, A., and Lloyd, S. Quantum algorithm for solving linear systems of equations. *Physics Review Letter* 15(150502). 2009.

[5] Lloyd, S., Mohseni, M., and Rebentrost, P. Quantum principal component analysis. *Nature Physics* 10, 9 (2014).

[6] Rebentrost, P., Mahsoud, M., and Lloyd, S. Quantum support vector machine for big data classification. *Physics Review Letter* 113(130503). 2014.

[7] Childs, A. M. Quantum algorithms: equation solving by simulation. *Nature Physics* 5, 12 (2009).

[8] Wiebe, N., Granade, C., Ferrie, C., and Cory, D. G. Hamiltonian learning and certification using quantum resources. *Physical Review Letters*. 112 (190501). 2014.

[9] Bisio, A., Chiribella, G., D'Ariano, G. M., Facchini, S., and Perinotti, P. Optimal quantum learning of a unitary transformation. *Physical Review A*. 81(032324). 2010.

Biography

Bingjie Wang joined the quantum computing scene as part of the 2011 Quantum Cryptography School for Young Students at the University of Waterloo. He earned a B.A. from the University of Cambridge with a thesis on machine learning for quantum entanglement detection algorithms.

1528-4972/16/09 $15.00

Many-body Quantum Mechanics: Too big to fail?

Special purpose quantum computers—realized with current technology—have the potential to revolutionize physics, chemistry, and materials science.

By Michael L. Wall, Arghavan Safavi-Naini, and Martin Gärttner

DOI: 10.1145/2983537

Quantum computers, devices whose components obey the laws of quantum mechanics, hold promise to solve problems that cannot be tackled with classical computers. As opposed to a classical computer, whose elements are bits that can take on two discrete values, a quantum computer is comprised of quantum bits, or "qubits," which are quantum mechanical objects with two distinct states. The essential advantage of qubits over bits is they display the quantum mechanical features of "superposition"—a qubit is in a linear combination of its two states—and "entanglement"—the outcome of measurements on one qubit is perfectly correlated with another qubit (see Figure 1), even though measurements on either qubit alone give random values. Over the past three decades, enormous effort has been put toward the construction of a universal quantum computer with some fantastic successes. Still, at present, realistic quantum computing remains in its early stages. The most fruitful application of quantum computation might turn out to be a more specialized one, which has the potential to revolutionize research in condensed matter physics, materials science, and chemistry. The prototypes of which have already been realized in labs around the world as the quantum simulator.

The very notion that there are some tasks quantum computers can perform efficiently and classical computers cannot means the computational complexity, class-bounded error quantum polynomial time (BQP), consisting of problems that are tractable (solvable in polynomial time) on a quantum computer, is assumed to be different than the complexity class *P*, which contains problems that are tractable on classical computers. Perhaps the most prominent example of a problem that is in BQP but thought not to be in *P*, is

integer factoring: Given an integer N, what are its prime factors? The best known classical algorithm for solving this problem scales in sub-exponential time, but the now-famous quantum algorithm discovered by Peter Shor solves this same problem in polynomial time [1].

Many widely used cryptographic schemes rely on the fact that it is hard for classical computers to factor large numbers. That a large-scale quantum computer can efficiently break these schemes has been a powerful impetus driving the field of quantum computing forward. In spite of this, there also exist so-called post-quantum cryptography schemes—tough even for quantum computers to crack [2]. Hence, while integer factoring is a fascinating example of the power of quantum computers, it is likely not the "killer app" of quantum computation. Controllable quantum many-body systems consisting of a few tens or more particles, even while not capable of universal quantum computation, can efficiently perform tasks that are intractable on classical computers; namely, solving for the dynamics of quantum many-body systems. This seemingly tautological insight, first put forward by Richard Feynman [3], has led to the field of quantum simulation. In which "designer" quantum many-body systems built of well-characterized components—such as ions, atoms, molecules, and super-conducting electrodes at very low temperatures—realize the dynamics of some other quantum system of interest—such as a high temperature superconductor, which is much harder to control or probe experimentally.

Before discussing quantum simulation in more detail, it is worth pointing out the enormous conceptual difference between the classical and quantum many-body problems. In classical mechanics, our "microscopic" degrees of freedom are the positions and momenta of N particles, which, in three dimensions, is a set of $6N$ real numbers. Given a set of $6N$ initial conditions for these variables, one then simulates dynamics by solving $6N$ (coupled) differential equations, e.g. Newton's equations. If our particles interact only pairwise, as is essentially always the case, each one of the differential equations involves N terms, and our total computation scales as $O(N^2)$. Hence, the classical many-body problem can be solved in poly(N) time on a classical computer, and so is in the complexity class P. This class of prob-

Figure 1. (a) Classical bits are binary, while qubits can be in any superposition state defined by two coefficients that can be represented as the surface of a sphere. (b) The number of coefficients needed to represent N qubits scales exponentially, which makes quantum problems prohibitively hard to solve on classical computers. (c) Two qubits can be in an entangled state.

The example given here shows perfect anti-correlations in the outcome of a measurement performed on both qubits. Such entanglement is the resource that enables quantum computers and simulators to outperform classical computers.

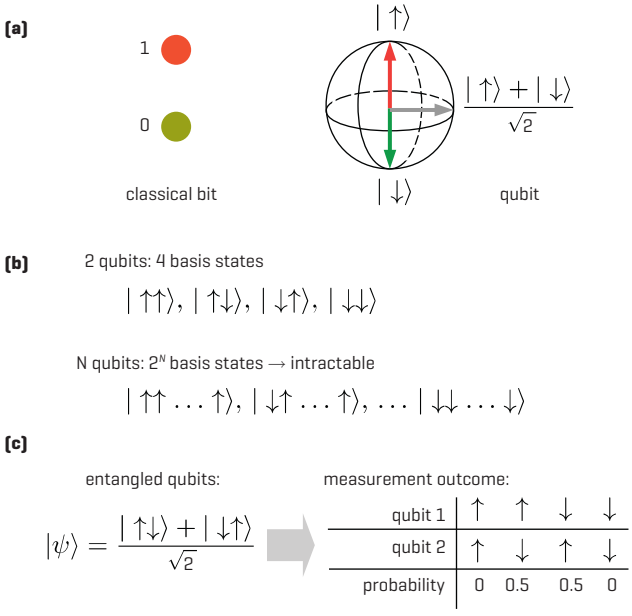


Image by Kotoffei/Shutterstock.com

lems essentially defines what it means for a problem to be tractable.

It should be noted that for a macroscopically large system, say on the order of Avogadro's number of particles ($N \sim 10^{23}$), we still cannot efficiently simulate the dynamics of every single particle efficiently by solving the coupled differential equations as outlined previously. However, for such a large system, the absolute positions and momenta of all particles is completely useless information for obtaining the macroscopic-scale behavior of the system that we usually care about. In addition, macroscopically large systems will also be subject to deterministic chaos, which makes the specification of all 10^{23} initial conditions irrelevant. In such situations, it is more useful to "coarse grain" the system and ask, not about the behavior of its microscopic constituents, but about macroscopic observables. This is in the spirit of defining thermodynamic observables, such as pressure or temperature for a gas, which describe the state of the macroscopic system, but contain no information about the microscopic properties.

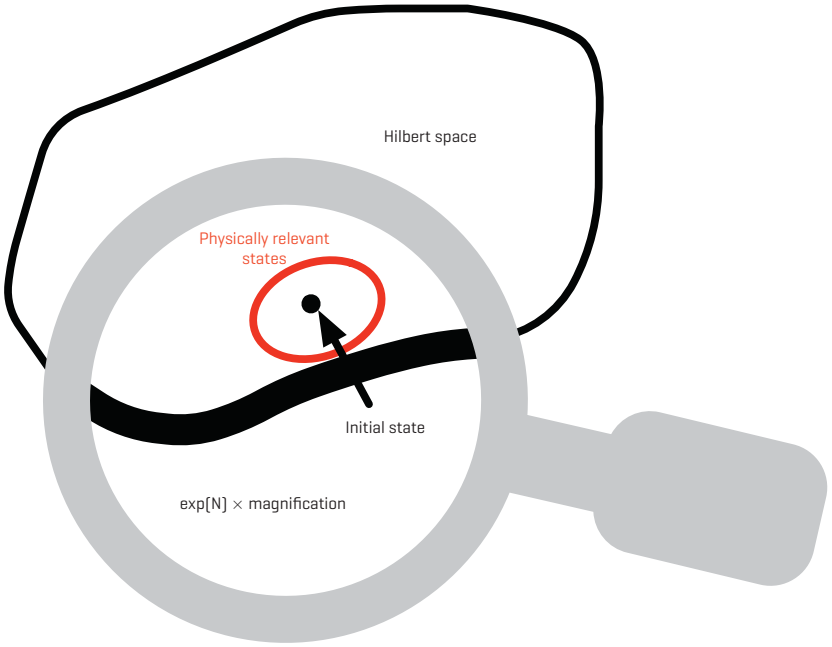
Now we turn to quantum problems. For concreteness, we will consider quantum problems, which involve some particles arranged in a regular array and each particle has a few discrete quantum states it can occupy. Physicists often call these objects "spins," in analogy to the discrete spin degree of freedom of elementary particles such as electrons. In addition, we will further specialize to the case of "spin-1/2," in which each particle has exactly two states, which we will denote $|\uparrow\rangle$ and $|\downarrow\rangle$. Such elemental objects can also be thought of as the qubits defined previously. Here, $|\bullet\rangle$ denotes a vector in Hilbert space, the normed linear vector space in which quantum mechanics is formulated. For a single spin, its state $|\psi\rangle$ is completely specified as a linear combination of the two basis states above, $|\psi\rangle = a|\uparrow\rangle + b|\downarrow\rangle$, where, e.g., $|a|^2$ can be interpreted as the probability to measure the system to be in the state $|\uparrow\rangle$. The dynamics of this state is described by the Schrödinger equation (using units where $\hbar=1$)

$$i\frac{\partial}{\partial t}|\psi\rangle = \hat{H}|\psi\rangle$$

where $\hat{H}$ is a matrix acting on Hilbert space, called the Hamiltonian, which represents the total energy of the system. Hence for a single spin, solving the Schrödinger equation amounts to integrating a 2 x 2 matrix differential equation—this is easy to do.

What happens as we increase the number of spins in our array? It is insightful to first consider the case of two

Figure 2. The set of states that a reasonable quantum system can reach in a reasonable time is an exponentially small fraction of all allowed states.



spins [1]. Here, the postulates of quantum mechanics specify the Hilbert space of two particles is spanned by the four states: $|\uparrow\uparrow\rangle$, $|\uparrow\downarrow\rangle$, $|\downarrow\uparrow\rangle$, and $|\downarrow\downarrow\rangle$, where the left and right objects in $|\bullet\rangle$ denote the quantum state of the two particles in some ordering (say, the left and the right particle in a one-dimensional array). Suppose we are now given the state of the system as: $a|\uparrow\downarrow\rangle + b|\downarrow\uparrow\rangle$ with $a = b = 1/\sqrt{2}$. We are asked what are the states of particle 1 and particle 2? Using our prescription , we find $|a|^2 = 1/2$ of the time we measure particle 1 to be in the state $|\uparrow\rangle$, and $|b|^2 = 1/2$ of the time we measure particle 1 to be in $|\downarrow\rangle$, and similarly for particle 2. However, strikingly, we note whenever we measure particle 1 to be in $|\uparrow\rangle$ we always measure particle 2 to be in $|\downarrow\rangle$, and similarly with $|\uparrow\rangle$ and $|\downarrow\rangle$ reversed. This state illustrates the quantum phenomenon of entanglement mentioned earlier, in which the behavior of two quantum objects are perfectly correlated with each other even though each object displays some randomness. Let's pause to consider how very strange this entangled state is: It is as if we had two double-sided coins that we flipped at the same time. We measure coin 1 to be up half of the time and down half of the time, but we always measure coin 2 to be the opposite of coin 1.

The two particle situation gives us the first inkling of how entanglement produces strange non-classical effects, but something even more profound occurs when we consider larger numbers of spins N. Namely, for N spins the complete realm of quantum possibilities is given by a linear combination of the states $|\sigma_1\ \sigma_2 ... \sigma_N\rangle$, where each of the σ_i can be $\uparrow$ or $\downarrow$. Counting all these possibilities, we see the size of the Hilbert space for this system, which is the "arena" that quantum mechanics lives in, grows exponentially with the number of spins. Hence, obtaining the dynamics of N particles (or even storing a quantum state of as many particles), is a problem that could be solved with poly(N) resources in classical mechanics, but requires exp(N) resources quantum mechanically. Let's again pause to consider how disastrous this is for solving quantum dynamics on a classical computer. For the sake of argument let's say we had at our disposal as many classical bits as we estimate there are particles in the visible universe, $\sim 10^{80}$. With this universe-sized classical computer, we could still only store the quantum state of about 260 spins. Clearly, simulating the behavior of a macroscopically-sized array of quantum particles through this approach is doomed to fail.

One could reasonably ask the question: "Why is the arena of many-body quantum mechanics so vast?" The answer is entanglement; a state pulled at random from Hilbert space is generally very highly entangled, displaying nonlocal correlations between measurement outcomes for all of its constituent particles. Again using our coin toss analogy, a "typical" many-body quantum state in Hilbert space would correspond to finding random outcomes for each coin when a million coins are tossed, but these random outcomes are near-perfectly correlated when all coins are considered collectively. Such very highly entangled states are quite fragile in the sense that their correlations are destroyed when placed in a noisy classical environment.

A natural question, then, is whether an experiment with a quantum many-body system utilizes the full Hilbert space?

The physical answer to this question is no! This can be made precise by asking how far in Hilbert space can a given quantum state travel after a reasonable evolution by the Schrödinger equation with a reasonable Hamiltonian operator $\hat{H}$. Here, a reasonable evolution means we only allow the evolution to occur for a time that scales polynomially with the number

Over the past three decades, enormous effort has been put toward the construction of a universal quantum computer with some fantastic successes.

of particles, and a reasonable Hamiltonian is one in which interactions occur between a fixed number of particles at a time (as occurs for all "natural" interactions like electromagnetic forces). In fact, the "volume" of Hilbert space that can be accessed with such a reasonable experiment is exponentially tiny (see Figure 2)[4]. Hence, the Hilbert space really is too big to fail. In order for quantum mechanics to be complete mathematically, we must include all of the highly entangled states that occupy the vast volume of a many-body Hilbert space. But most of these states simply have no meaning physically, meaning that it is difficult even for a physical quantum computer to prepare them.

While the previously stated result seems to preclude useful computation with quantum many-body systems, it also points out one specific problem that reasonable quantum systems can solve efficiently in a reasonable time, which is generating the dynamics of reasonable quantum many-body systems over reasonable times. This statement is not vacuous, as we do not know how to simulate generic, yet reasonable, quantum many-body systems with classical computers.

Over the last decades physicists have vastly improved their abilities to cool atoms to temperatures near absolute zero, to suppress any interaction with their (classical) environment, and to control and tune the interactions between them. These developments enabled them, using various different physical set-ups, to get remarkably close to the realization of quantum simulators with near-perfect isolation and exquisite control over all degrees of freedom. Milestone examples range from atoms and molecules trapped in lattices formed by laser beams, charged atomic ions in self-assembled structures, and arrays of superconducting electrodes coupled through Josephson junctions [5]. Among the most exciting implementations are fermionic alkali atoms, such as lithium, cooled to millionths of a degree above absolute zero and trapped in crystals of light mimicking a solid-state ionic lattice. Their interactions can be tuned so they behave like electrons in an unconventional superconductor, potentially providing insights into problems from condensed matter physics that are prohibitively hard to solve using a classical computer. Solving quantum many-body problems is not only of interest to basic research in physics, such problems are also central to chemistry and materials physics. Enormous amounts of supercomputer time are spent studying quantum many-body phenomena. Hence, the development of a non-universal quantum computer that could efficiently solve for chemical or material structures would completely revolutionize the development of new technologies.

Many widely used cryptographic schemes rely on the fact that it is hard for classical computers to factor large numbers.

So, what is left for current theorists, who at present only have access to classical computers, to do? For one, emerging quantum simulators must be verified to perform as we think they should in benchmark experiments. While this is a very difficult problem, there are certain places in which physicists have developed well-controlled approximations that can be used to compare with quantum simulators. For example, in one spatial dimension, our understanding of the entanglement structure of the exponentially tiny fraction of Hilbert space that we can access has led to a framework known as matrix product states, which enable efficient computations over moderate times. For certain systems in two and higher dimensions, we can estimate equilibrium properties of certain quantum many-body systems by treating them as classical systems in one higher dimension and sampling their trajectories with probabilistic Monte Carlo schemes. The direct comparison of such advanced numerical techniques with quantum simulation experiments not only gives experimentalists faith that the simulator is performing as it should, but also gives theorists a unique opportunity to test and hone new methods. This positive feedback between theory and experiment has become a hallmark of the burgeoning quantum simulation field.

In conclusion, we argue perhaps the most useful enterprise a quantum computer can do is to forget about being a computer and just behave as a quantum many-body system. Even though such a many-body system can only explore a tiny fraction of the realm of mathematical possibilities afforded to it by the framework of quantum mechanics, this tiny fraction encompasses essentially all systems of interest from physics, chemistry, and materials science. Using designer quantum systems to simulate many-body physics holds great promise for teaching us about the structure of our quantum world, and has the potential to transform computation as we know it.

References

[1] A CS-accessible account of quantum computing may be found in: Nielsen, M. and Chuang, I. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2000.

[2] See, e.g., Chen, L. *et al.* Report on post-quantum cryptography. National Institute of Standards and Technology Internal Report 8105. 2016.

[3] Feynman, R. P. Simulating Physics with Computers. *International Journal of Theoretical Physics* 21 (1982), 467.

[4] Poulin, D. *et al.* Quantum Simulation of Time-Dependent Hamiltonians and the Convenient Illusion of Hilbert space. *Physical Review Letters* 106 (2011), 170501.

[5] Cirac, J. I. and Zoller, P. Goals and opportunities in quantum simulation. *Nature Physics* 8 (2012), 264.

Biographies

Michael L. Wall is a senior research associate at JILA in Boulder, CO. His Ph.D. work at the Colorado School of Mines earned him the Nicholas Metropolis award for computational physics from the American Physical Society. His present research focuses on verifying and applying emerging quantum technologies in atomic, molecular, and optical systems, and on numerical methods for quantum many-body problems.

Arghavan Safavi-Naini is a research associate at JILA in Boulder, CO. She received her Ph.D. from the Massachusetts Institute of Technology. She develops numerical techniques to characterize equilibrium and non-equilibrium properties of quantum many-body systems with long-range interactions, such as those realized by lattice gases, polar molecules, as well as trapped ion systems.

Martin Gärttner is a research associate at JILA in Boulder, CO. He received his Ph.D. from the University of Heidelberg, Germany. His research focuses on the non-equilibrium dynamics of strongly interacting quantum systems, such as trapped ions, polar molecules, and Rydberg atoms.

Black Holes, Quantum Mechanics, and the Limits of Polynomial-Time Computability

Which computational problems can be solved in polynomial-time and which cannot? Though seemingly technical, this question has wide-ranging implications and brings us to the heart of both theoretical computer science and modern physics.

By Stephen P. Jordan

DOI: 10.1145/2983539

The fundamental limits of computation have long been of interest to computer scientists, physicists, and even philosophers. The roots of computer science lie in early 20th century investigations by Kurt Godel, Alan Turing, Alonzo Church, and others to determine what problems are solvable by computation, and what problems are fundamentally and permanently undecidable.

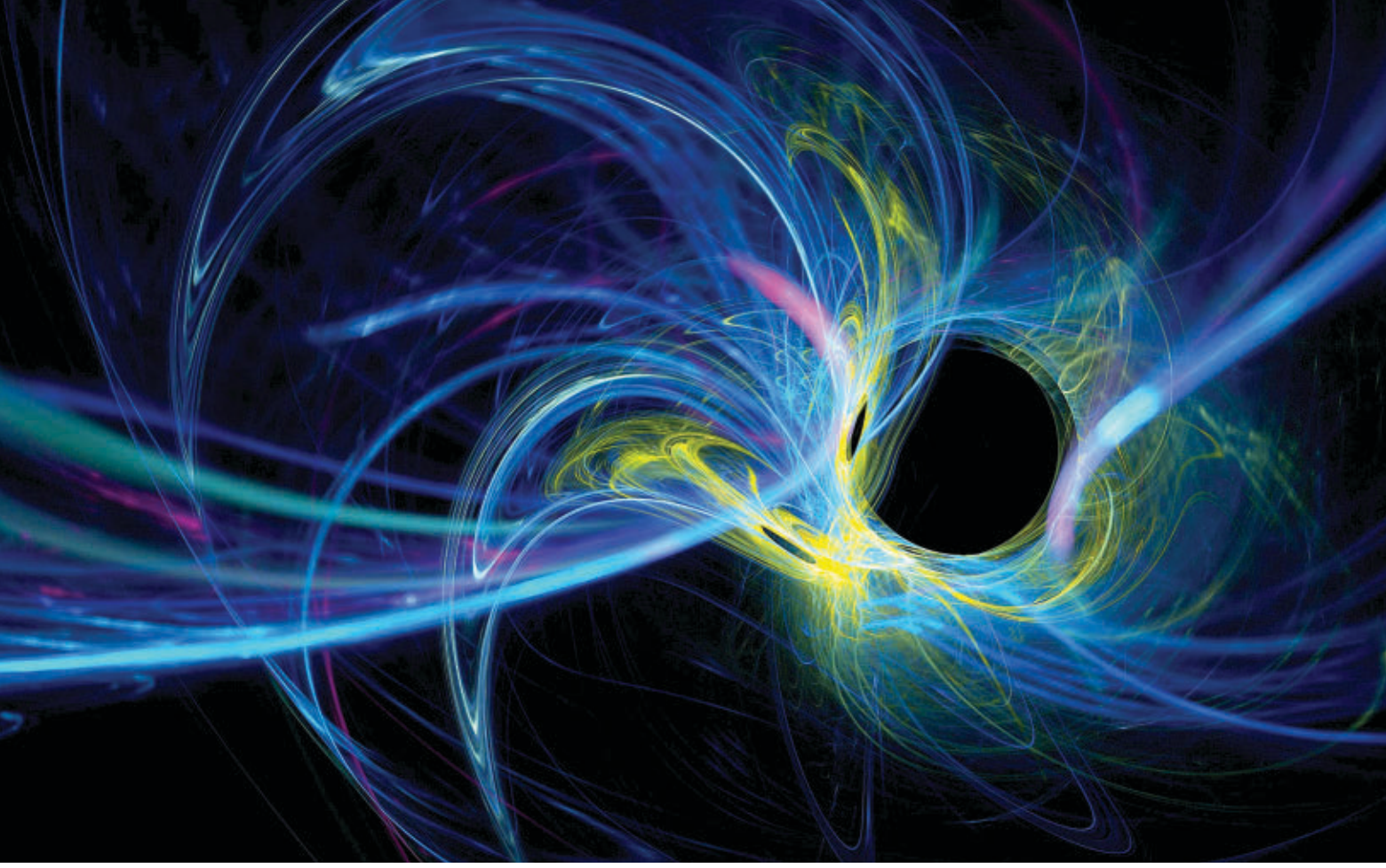
Modern computer science focuses on a more fine-grained question: Which problems can be solved efficiently and which are intractable? Over the years a consensus has emerged; the most fruitful mathematical formalization of efficiency is that problems be solvable using a number of computational steps that scales polynomially with the size of the problem, measured in the number of bits needed to describe the problem instance. We call the set of problems answerable by a standard computer (Turing machine) using polynomially many steps complexity class P (short for polynomial-time). Until recently, it was thought P was the final word on what class of computational problems can be efficiently solved.

Instead of a Turing machine, one may replace the underlying model of computation with uniform families of logic circuits, parallel computers such as Turing machines with polynomially many tapes, or even abstract systems of symbol replacement such as Church's lambda calculus. In fact, many plausible definitions of polynomial-time computability all turn out to yield P as the resulting set of solvable problems, and in each case, the set of problems solvable in polynomial time remains the same. Although a problem requiring n^2 time to solve in one model may, for example, require n^3 time in another.

The insensitivity of polynomial-time computability to the details of the underlying model of computation led to the following conjecture, known as the complexity-theoretic Church-Turing thesis: *"A Turing machine can simulate any realistic model of computation with polynomial overhead."*

The vague nature of this statement has long been a source of discomfort. What does "realistic model of computation" really mean? One proposal for sharpening this statement, thanks to David Deutsch, is to interpret "realistic" as "physically realistic." The real goal is to classify the power of computational devices that are allowed by the laws of physics and use a polynomial amount of space, time, and energy.

For many years, Deutch's interpretation of the complexity-theoretic Church-Turing thesis held up remarkably well. Realistic alternative models of digital computation were consistently found to be efficiently simulatable by standard Turing machines. Models of computation that appeared to yield exponentially greater computational power than Turing machines always turned out to be unrealistic. Most of those models were forms of analog computation, which, when examined carefully, turned out



to depend on operations with exponentially high precision. When realistic errors were taken into account, the apparent exponential advantages over Turing machines fell away.

All of this dramatically changed when the concept of quantum computation was discovered. In the 1980s, Richard Feynman proposed certain quantum systems of many particles apparently take exponential time to simulate with conventional, classical computers. Feynman suggested these systems may be fundamentally impossible to simulate in polynomial time by conventional Turing machines, and therefore by harnessing these systems, one may be able to perform certain computations that are intractable on conventional Turing machines. To mathematically capture the computational power of quantum systems, various formal models of universal quantum computation were defined, including quantum Turing machines and quantum logic circuits. Subsequently, these models were all proven to simulate one another with polynomial overhead. The complexity class of problems solvable in polynomial-time by any of these models is the same, and is called BQP (which stands for Bounded-error Quantum Polynomial-time). Based on this, Deutsch argued the complexity-theoretic Church-Turing thesis was now defunct and must be replaced by a new quantum Church-Turing thesis: *"A quantum Turing machine can efficiently simulate any realistic model of computation with polynomial overhead."*

However, two significant objections remained in the minds of skeptics, who believed the original Church-Turing thesis was in no need of a replacement. First, is a quantum Turing machine truly a realistic model of computation? Maybe it somehow depends on unrealistic assumptions about exponentially precise operations, much like earlier unrealistic models of analog computation. Second, is it really true that quantum Turing machines cannot be simulated efficiently by classical Turing machines? Maybe the difficulty of simulating quantum systems is not fundamental, but simply a failure to find the right algorithm. Although neither of these objections have been answered definitively, developments over the last few decades have greatly bolstered the case that the leading candidate for the set of problems in-principle solvable with polynomial resources in our universe is BQP, and not *P*.

To definitively refute the objection that quantum computers are not realistic would require building a scalable universal quantum computer. Despite substantial effort and remarkable progress, this has not yet been achieved. However, there is now strong theoretical evidence that quantum computing is fundamentally different from the models of analog computation that previously gave false hope of solving problems outside of *P*. Fault-tolerance threshold theorems, first proven in the mid 1990s, show once the error rate per logic gate of a quantum computer is brought below a fixed finite threshold, arbitrarily long quantum computations can be carried out reliably through the use of error-correcting codes. Proving the power of quantum computation is not simply an illusion arising from unrealistic modelling of error. More concretely, threshold theorems serve as blueprints for building large-scale quantum computers once experimental devices reach the necessary precision thresholds, a milestone that some recent experiments are claimed to have met.

To refute the objection that quantum computers might be efficiently simulated by classical computers, one would have to prove that some problem quantum computers can efficiently solve is outside of *P*. Proving this rigorously is an extraordinarily difficult mathematical problem, which has not been cracked. However, polynomial-time quantum algorithms have been discovered for several problems that are widely believed to lie outside of *P*. Most famously, in 1994, Peter Shor discovered polynomial-time quantum algorithms for integer factorization and discrete logarithms. The belief that these problems cannot be solved by any polynomial-time classical algorithm is the foundation of the public-key cryptosystems used to protect financial transactions on the internet.

The quantum Church-Turing thesis is a bold claim about fundamental science. In answer to the question "What is the computational power of the universe?" the quantum Church-Turing thesis replies: "The set of problems efficiently solvable in our universe is BQP." One of the most exciting and fundamental questions in either physics or computer science is whether this is really true. In light of the fault-tolerance threshold theorems proven in the 1990s, it seems unlikely, in principle, the set of problems solvable in polynomial time is smaller than BQP. The essential challenge to the quantum Church-Turing thesis is therefore the possibility of exotic physical systems that are intractable to simulate even with a quantum computer.

It is possible that exotic new physics will be discovered by future experiments, which implies computational power far beyond that of "ordinary" quantum computers. However, even without such discoveries, there is already a job for theorists to do to probe the quantum Church-Turing thesis. At present, it is not clear that efficient quantum algorithms can simulate all of known physics, so we must try to find them. Success in this endeavor yields quantum algorithms that will be useful once quantum computers are built. On the other hand, "failure" is even more interesting! If we find some physical system is intractable to simulate even for quantum computers, then this challenges our fundamental assumptions about polynomial-time computability, as enshrined in the quantum Church-Turing thesis. Furthermore, such physical systems might eventually be harnessed to perform computations intractable even on quantum computers.

In the 1980s, Richard Feynman proposed certain quantum systems of many particles apparently take exponential time to simulate with conventional, classical computers.

At present, theoretical work on quantum algorithms for simulating physical systems is at a fairly advanced stage. Polynomial-time quantum algorithms of substantial generality have been discovered that can simulate chemistry and materials science. The remaining challenges to the quantum Church-Turing thesis come from physical systems in which quantum effects and relativistic effects are both significant.

To describe the behavior of systems travelling at significant fractions of the speed of light, physicists turn to a branch of physics called special relativity, which is known to most of us through Einstein's predictions about relativistic time-dilations for space travelers moving at close to the speed of light and his famous formula, $E = mc^2$. However, special relativity is not only relevant to science-fiction scenarios involving futuristic spacecraft. Many experimentally accessible systems simultaneously involve speeds high enough and objects small enough that special relativity and quantum effects, respectively, cannot be neglected. Examples include radioactive decay, nuclear reactions, cosmic rays, and particle collisions in accelerators such as the Large Hadron Collider. In fact, if carried to sufficiently high precision, even observations of the spectral lines of atoms—of the sort carried out in many high-school science labs—yield results that can only be predicted by taking into account both quantum and relativistic effects.

The theory that successfully unifies quantum mechanics with special relativity is called quantum field theory. By the late 1970s, a single quantum field theory, called the "Standard Model," describing all known particles and forces with the exception of gravity, had been proposed. Some aspects of the Standard Model (relating to the electron's magnetic moment) have shown agreement between experiment and theory to 10 digits of precision, making them amongst the most precisely tested predictions in all of science. With the observation of the Higgs boson by Large Hadron Collider, all particles in the Standard Model have now been observed.

A crucial test of the quantum Church-Turing thesis is therefore whether polynomial-time quantum algorithms can simulate the Standard Model. Because the Standard Model describes all known physics other than gravity, including quantum computers being developed in labs around the world, it is believed classical computers cannot efficiently simulate the Standard Model. If they could, then this would enable them, for example, to efficiently factor large integers by simulating a quantum computer executing Shor's algorithm.

Given the success of polynomial-time quantum algorithms for simulating non-relativistic quantum systems, the obvious first question to ask is whether simulating the Standard Model should be any different. Do we get efficient algorithms for simulating quantum field theories as easy corollaries of previously known quantum algorithms for simulating quantum systems? In fact, we don't. The simulation of quantum field theories poses new challenges. Perhaps the most fundamental of these challenges is that quantum field theories involve infinitely many degrees of freedom, even within a finite volume.

As one might guess, the infinite degrees of freedom in quantum field theory lead to many thorny problems, both mathematically and computationally. Nevertheless, recent work has demonstrated, at least for simple examples of quantum field theories, efficient quantum algorithms for simulation can be obtained through careful discretization [1]. Much work remains to be done to

adapt these quantum algorithms to handle more complex quantum field theories such as the Standard Model. Nevertheless, it appears likely the Standard Model can be efficiently simulated using generalizations of the techniques in [1].

If polynomial-time quantum algorithms to simulate the Standard Model are found, we will still be left with one final frontier in the quest to test the quantum Church-Turing thesis: quantum gravity. Einstein's theory of general relativity describes gravity as curvature of spacetime. The predictions of general relativity have been confirmed by many astronomical observations, most recently the direct detection of gravitational waves. However, general relativity is not a quantum theory. It is not known how to model gravity in regimes where quantum effects cannot be neglected, such as in black holes and the Big Bang. A number of candidate theories describing quantum gravity have been proposed, with string theory the most well-known.

Can quantum gravity be efficiently simulated by quantum computers? This question is extremely challenging, not least of which because a complete theory of quantum gravity is not yet known. In the absence of a complete theory of quantum gravity, how can the problem of simulating quantum gravity be addressed? One approach is to use leading theoretical models of quantum gravity such as string theory. At present, this research frontier is largely unexplored, with the exception of quantum algorithms to simulate some topological quantum field theories (TQFTs) and a related class of models called spin-foams. Encouragingly, these quantum algorithms have proven useful independently of whether these models are successful in the context of quantum gravity. TQFTs are now thought to describe the emergent behavior of certain materials observed in low-temperature physics laboratories. Additionally, the simulations of TQFTs and spin-foams have yielded, as a byproduct, efficient quantum algorithms to approximate topological invariants that are of interest independent of their connection to physics.

A second approach to understanding the complexity theory of quantum gravity is to take a "phenomenological" point of view. That is, even lacking a fundamental theory of quantum gravity, certain broad features of the theory may be guessed from symmetry principles and other general arguments. The history of physics contains many examples of systems for which accurate quantitative predictions were extracted from phenomenological models long before the underlying microscopic theory was fully understood. Ideal gasses, low-temperature superconductors, and black-body radiation are three prominent examples.

Black holes are a class of physical systems in which quantum-gravitational effects are expected to be important, and have been studied extensively from a phenomenological point of view. In the case of black holes, the arguments that can be made by applying various well-established symmetries and physical principles seem to yield results that are in conflict. The most recent version of this conflict is called the "Black Hole Firewalls Paradox," which descends from a long line of arguments about whether information is destroyed after falling into black holes.

Some phenomenological models of black holes, which are currently being debated in the context of the firewalls paradox, contain features that, at least at first glance, appear to have shocking consequences for polynomial-time computability. As an example, in the final-state projection model of Horowitz and Maldecena, the quantum states of black holes may evolve nonlinearly in time. Generically, nonlinear transformations of quantum states can be used to obtain polynomial-time solutions to *NP*-hard problems, as was earlier shown by Abrams and Lloyd. Not even quantum computers are believed to solve *NP*-hard problems in polynomial time. A physical device achieving this would be shockingly powerful. For any problem whose solution can be verified efficiently, such a device would efficiently find a solution. In particular, algorithms for verifying formal mathematical proofs could be bootstrapped to yield efficient algorithms finding proofs, making human mathematicians obsolete [2].

Could black holes be carrying out computations exponentially more complex than those allowed by our current theories of quantum computation? Recent analysis throws some cold water on this provocative suggestion [3]. In examining several models of black holes for which quantum mechanical principles are modified, it is found if the models are pushed into parameter regimes that imply efficient solution to *NP*-hard problems, they also imply the ability to send superluminal signals using quantum entanglement. According to special relativity such superluminal signaling allows messages to be sent backward in time, leading to numerous logical paradoxes. Thus, most physicists believe in a principle called "causality," which states that superluminal signaling is impossible and its presence in a theory is a sign the theory is incomplete, wrong, or pushed beyond its limits of applicability.

Physical arguments cannot resolve the mathematical problems of complexity theory, such as proving that neither *P* nor BQP contains NP. However, taking for granted that these standard assumptions about complexity theory are true, one is left with an interesting physical question: Which complexity class corresponds to the problems solvable with polynomial resources in our universe? The results suggest the impossibility of solving *NP*-hard problems in polynomial time by physical means is a robust consequence of basic physical principles such as causality, rather than a fragile inference dependent on all of the precise details of quantum mechanics [3]. The quantum Church-Turing thesis once again comes out looking strong. Testing it further remains an outstanding open problem and a grand adventure promising to yield new insights into both physics and computer science.

References

[1] Jordan, S.P., Lee, K.S.M. and Preskill, J. *Quantum algorithms for quantum field theories. Science* 336, 6085(2012), 1130–1133. arXiv:1111.3633.

[2] Fortnow, L. What if P = NP?. Computational Complexity. Blog. May 25, 2004; http://blog.computationalcomplexity.org/2004/05/what-if-p-np.html

[3] Bao, N. Bouland, A and Jordan, S.P. *Grover search and the no-signaling principle.* 2015. arXiv:1511.00657.

Biography

Stephen Jordan is a physicist at the National Institute of Standards and Technology (NIST), a fellow of the Joint Center for Quantum Information and Computer Science (QuICS), and adjunct associate professor of computer science at the University of Maryland. He is also the author and maintainer of the "Quantum Algorithm Zoo," a comprehensive online reference on quantum algorithms.

Reliable Quantum Circuits Have Defects

The first large-scale practical quantum computer is within reach. Coming to grips with the strategy and challenges of preparing reliable executions of an arbitrary quantum computation is not difficult. In fact, defects are good.

By Alexandru Paler, Austin G. Fowler, and Robert Wille

DOI: 10.1145/2983541

In the not too distant future, a quantum computer engineer will be confronted with the problem of automating the compilation of what a user wishes to execute (quantum algorithms) to instructions that a quantum computer is able to execute.

A quantum algorithm is implemented as a quantum circuit formed of quantum gates operating on quantum bits (qubits). Executing a quantum circuit is different compared to classical circuit execution. The wires and the gates of a classical circuit are implemented in hardware. In contrast, for a quantum circuit only the qubits may be seen as part of the hardware, because the gates are understood as instructions for transforming qubit states. There are multiple related gate sets that can be used to express quantum algorithms, and their relation is similar to how classical programming languages are compiled from high-level ones into a lower level and finally assembler instructions.

To solve the problem, our quantum computer engineer should learn about how defects are useful for constructing reliable quantum circuits [1]. Then devise a method to automatically transform the high-level description of the quantum algorithm into an equivalent low-level description. This would be straightforward in an ideal world, but in reality, quantum hardware is faulty. State of the art quantum computing architectures are founded on the decision to use scalable, but faulty, quantum hardware in conjunction with an efficient error correcting code capable of tolerating high error rates. The solution is to chose a suitable quantum error correcting code and to compile the algorithm into an intermediate description language, which guarantees very high computational reliability. The surface quantum error correcting code is chosen due to its excellent error correcting properties and very low resource overheads. The surface code can be used for hardware failing less than 1 percent of the time.

It is reasonable to assume the first quantum computer will be built from faulty hardware entities arranged in a two-dimensional lattice (see Figure 1a). Each entity in the lattice represents a physical qubit that can be manipulated individually or interacted with its nearest neighbors. Physical qubits can be either on (when actively manipulated) or off (when not being

Image by Erik Lucero, Courtesy of Martinis Group/UCSB

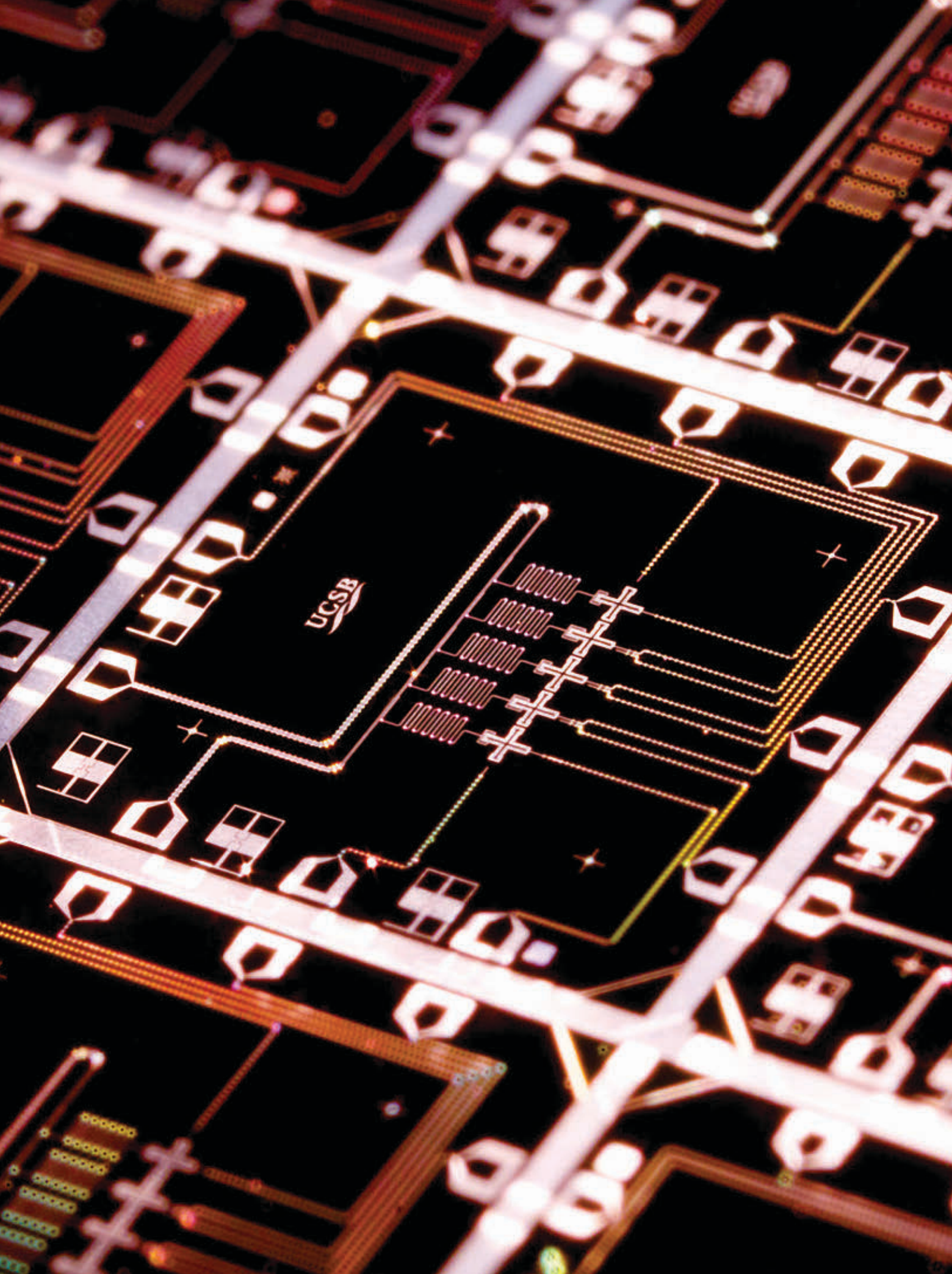

UCSB

Figure 1. From physical qubits to defects and braids: a) A lattice of 16 physical qubits, which can be switched off (green), is used in rounds by switching on primal qubits (blue) and dual qubits (red); b) The dual-primal CNOT is a single braid, and a primal-primal CNOT is three braids.

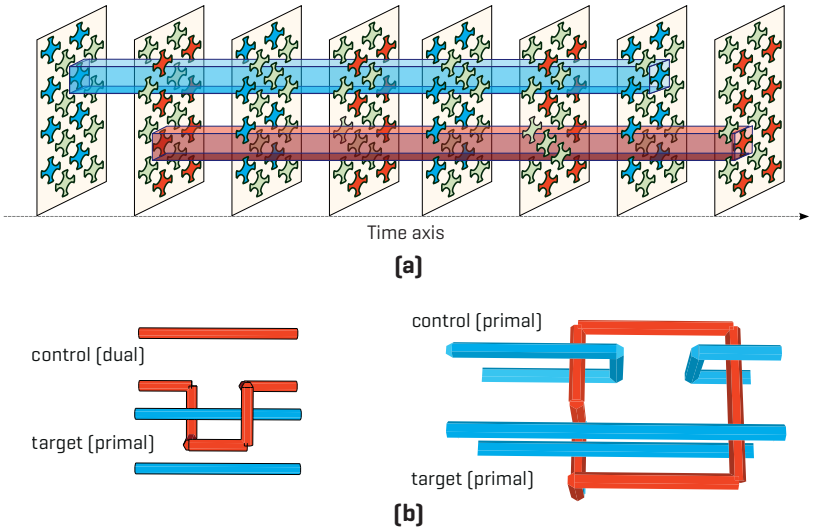


Figure 2. Quantum circuits. Horizontal wires abstract the manipulation of qubits. The CNOT gate is symbolized by the horizontal line connecting the · (control qubit) and the ⊕ (target qubit). Controlled measurements, indicated by double wires, determine the measurement basis of a qubit depending on the result of another qubit (e.g. the upper Z basis measurement). Following the circuits are ICM representations: a) The S gate; b) The T gate.

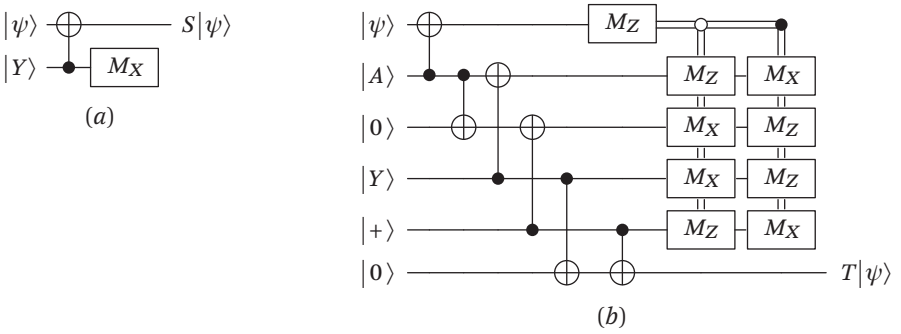
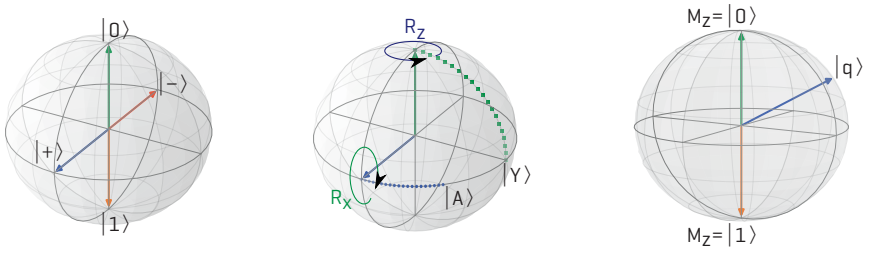


Figure 3. Visualization of state initialization, gate applications, and measurement. A qubit can take any state between north pole ($|0\rangle$) and south pole $|1\rangle$), (left). Single qubit gates are rotations around an axis, (middle). A measurement returns one of the two possible states, (right).



used). The available computational resources are constrained by the lattice area (number of physical qubits) and time (number of interaction rounds).

The engineer faces the challenge of compiling (as efficiently as possible with respect to the computational resources) an algorithm into surface code elements. Our hope is this article will alleviate the engineer's fear of defects, because these are basic error corrected elements used by the surface code.

QUANTUM CIRCUIT INGREDIENTS

Quantum circuits have their own particularities, given that they describe computations based on quantum physical effects. Firstly, a quantum circuit has the same number of inputs and outputs; secondly, all the gates have the same number of inputs and outputs; and thirdly, the state of arbitrary qubits cannot be copied. From a diagrammatic point of view, a quantum circuit is a set of horizontal wires interrupted by quantum gates (see Figure 2). Quantum circuit wires are qubit abstractions. The state of a qubit is transformed by each gate applied to that wire after a left-to-right traversal. Circuit inputs are on the left side, while the outputs on the right side.

It would be difficult to discuss circuits and computations without introducing a few technical details. The state of a qubit named q is a two-dimensional complex vector denoted $|q\rangle$ and imagined to indicate a point on the surface of a three-dimensional unit sphere (see Figure 3). The sphere poles are called computational basis states, and a state can also be on the equator in $|+\rangle$ or $|-\rangle$.

Quantum gates rotate qubit states around a sphere axis, and each single qubit gate can be decomposed into three rotations around two orthogonal axes, for example, Z and X. The bit flip transforming $|0\rangle$ into $|1\rangle$ is a π rotation around the X-axis. The $|+\rangle$ is transformed into $|A\rangle$ by a $\pi/4$ rotation around the Z-axis. Considering the arbitrary quantum gate G, we will write $G = R_Z(\alpha)R_X(\beta)R_Z(\gamma)$, where R_X and R_Z are rotation operators around the X and Z axes, and α, β, γ are rotation angles. This decomposition is a first example of how a high-level description (gate G) is compiled into a lower-level description (only rotation gates).

Single qubit measurements are probabilistic and performed around a sphere axis. The resulting state depends on where the axis touches the unit sphere surface; for example, a *Z*-basis measurement (symbolized by M_Z) returns either $|0\rangle$ or $|1\rangle$. The probability of the measurement result depends on the angle between the measured state and the measurement axis. Multi-qubit gates exist too. For exampl e, the CNOT gate is one possibility to create the quantum specific phenomenon of entanglement. Quantum entanglement occurs when pairs of qubits are interacted in a manner such that the state of each qubit cannot be described independently. The CNOT gate performs a bit flip of a computational state (target) if another state is $|1\rangle$ (control).

DEFECTS, BRAIDS, AND DISTILLATIONS

A bad environment also influences quantum circuits, resulting in faulty qubit initializations and measurements or faulty quantum gate applications. The majority of environment-induced faults can be mitigated by the quantum error correcting codes. From the perspective of the surface code, logical qubits are encoded and operated (initializations, measurements, and gates) by switching offsets of physical qubits in the lattice and interacting only the qubits that are still on. It suffices to mention there are two methods (primal and dual) for manipulating single physical qubits, and for saving the details about how the qubits interact with each other. Physical qubits, depending on their manipulation method, are switched on and off in turns: primal, dual, primal etc.

A defect abstracts how a set of switched-off lattice physical qubits is evolving in time, and depending on the physical qubits type, the defect can be either primal or dual. In Figure 1a, the primal defect (blue) abstracts the set of switched-off primal qubits (e.g. one qubit in two rounds), and the dual defect (red) abstracts switched-off dual qubits (e.g. one qubit in two rounds). A logical qubit is formed by pairs of same type defects and, as a result, the surface code allows the construction of primal and dual logical qubits.

A logical CNOT gate (see Figure 1b) is a braid between defects of opposite type (a primal and a dual): The dual logical qubit controls the primal logical qubit (target). Braids between defects of the same type leave the corresponding logical qubit states untransformed; the result is a logical identity gate. Braiding is the only straightforward operation between defects, implying arbitrary quantum circuits—consisting of logical qubit initializations, logical CNOT gates, and logical qubit measurements—can be easily protected by the surface code.

It is reasonable to assume the first quantum computer will be built from faulty hardware entities arranged in a two-dimensional lattice.

The error correction capability of the surface code (its distance) is a function of defect circumference and defect distances: construct distant, or thick defects when using very faulty hardware; and construct close, or thin defects when the hardware is less faulty. Code distance is not discussed herein because it does not influence the definition of defects and braids

The surface code will not solve all environment related issues. Although an initialized qubit will be protected against errors, it may have a low fidelity: Using the sphere visualization, there is a large distance between the actual state and the ideal state. Fidelity is increased by distillation procedures expressed as subcircuits [2]. These take multiple low fidelity instances of a state and output a single high fidelity state. Consequently, the surface code will have to protect circuits including distillation procedures.

THE RISC OF QUANTUM CIRCUITS

Reduced instruction set computing (RISC) was proposed as a way to increase classical computing performance, but the performance of quantum computers is not a thoroughly discussed research topic for the moment. However, there are sufficient reasons why a reduced set of quantum gates is useful. On the one hand, design automation methods can focus on a common framework; on the other, there are known efficient methods to actively protect specific gates against errors.

T is the Difficult Gate. Quantum computers seem more powerful for particular tasks that are exponentially difficult for classical computers. However there are exceptions. Take quantum computations using only Clifford gates; these are not universal and cannot express the full capabilities of quantum computing. Clifford gates are the Hadamard $H = R_Z(\pi/2)$ $R_X(\pi/2)R_Z(\pi/2)$, $V = R_X(\pi/2)$, $S = R_Z(\pi/2)$, the CNOT, and any other gate combination of the previous (e.g. *SHV*). The Clifford+*T* gate set is universal and indeed exponentially difficult for classic computers, because of the gate $T = R_Z(\pi/4)$.

All gate types are translatable into Clifford+*T*, and research has focused lately on this set. Computations, including state distillations, can be protected by the surface code if all the circuit's gates are decomposed into Clifford+*T* and then into an ICM representation (single qubit **i**nitializations, **C**NOT gates, and single qubit **m**easurements), the smallest set of gates [3, 4].

Initialize, Entangle, and Measure. ICMs are the RISC of surface code protected quantum circuits. Clifford+*T* gates are translated into subcircuits of only qubit initializations, CNOT gates, and qubit measurements. Initializations can be either one of four possible states: ($|0\rangle$, $|+\rangle$, $|A\rangle$, $|Y\rangle$). While measurements are of two types: single qubit independent ones (see Figure 2a) and classically controlled measurements (see Figure 2b).

An ICM single qubit rotation (e.g. *V*, *S*, *T*) about an angle ϑ is implemented by entangling an ancilla qubit (initialized into one of the four states) to the qubit representing the state to be rotated, and measuring one of the two qubits. Quantum measurements are probabilistic and will yield a rotation

Figure 4. An assembly of braided defects is obtained from an optimized circuit.

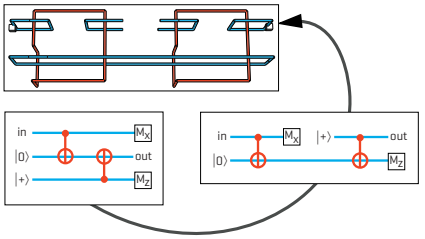
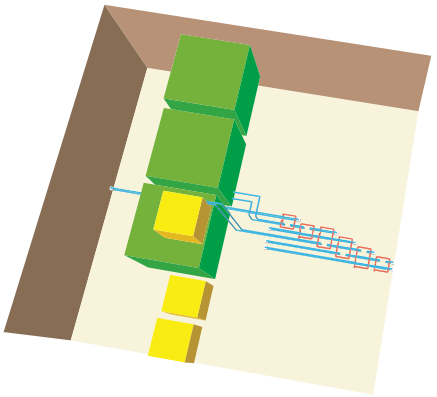


Figure 5. A defect assembly including distillation boxes (green and yellow). This is a resource-suboptimal placement of boxes because an imagined bounding box (orange) is largely unoccupied.



by ϑ or by $-\vartheta$ (rotation about ϑ in the opposite direction). The latter situation requires the wrongly rotated state to be corrected by an additional rotation of 2ϑ. For the V and S gates, the correction can be tracked through the CNOT circuit and does not need to be implemented as a gate. The ICM formulation of the S gate is illustrated in Figure 2a.

The ICM T gate ($R_Z(\pi/4)$) is slightly more complex because it requires an S gate correction ($R_Z(2\pi/4) = R_Z(\pi/2) = S$), which cannot be tracked. It needs to be executed inside the circuit and, therefore, four other ancilla qubits are entangled and measured depending on the topmost M_Z result. The circuit performs an $R_Z(\pi/4)$ rotation irrespective of the upper M_Z result.

"Forget About Small Efficiencies." Compiling a quantum circuit into surface code elements is halfway finished. The ICM form introduces a quantum circuit wire for each ancilla qubit, but not all the qubits are required simultaneously during a computation. Independent qubits can share the same wire so that a qubit measurement is not preceded by another qubit's initialization. This leads to less physical resources required for error correcting the computation.

ASSEMBLIES OF DEFECTS

At last, the quantum computer engineer can protect a quantum circuit against the bad environment. The circuit is translated into an assembly of braided defects. It is not specified if a qubit should be primal or dual. The solution is to consider all the qubits being primal, and implement a logical CNOT between primal qubits. Each CNOT from the circuit is implemented by the primal only braided logical CNOT from Figure 2b.

For example, in Figure 4 a circuit is optimized so two qubits share the same wire. The resulting logical qubits are pairs of blue (primal) defects, and each CNOT from the circuit is represented by a red (dual) defect braided three times around primal defects. The illustrated assembly shows the initialization and the measurement of a logical qubit depend on the existence of a third defect connecting defect pairs (e.g. compare $|0\rangle$ and $|+\rangle$ initializations)

The additional example from Figure 5 corresponds to the T gate from Figure 2b. The six logical qubits from the ICM circuit are the six parallel pairs of blue (primal) defects. Once more (similarly to Figure 4) each CNOT is implemented by a red (dual) defect braided three times. This example circuit includes distillations ($|A\rangle$ and $|Y\rangle$) symbolized by boxes of different volumes. Boxes are placeholders of surface code protected versions of the distillation subcircuits. Distillations are probabilistic (may not succeed and the output state has low fidelity) and heralded (it is known if distillation succeeded). The engineer computes the number of boxes to be sufficient for computational reliability, and lets all boxes execute in order to know which were successful and which not. Only outputs of successful boxes are usable. Figure 5 shows three distillations of each type that are executed (yellow and green), only the successful ones are connected to the circuit by defect pairs.

TO DO: BUILD A QUANTUM COMPUTER

The T gate, which makes quantum computations difficult to simulate classically, also largely dictates defect placement. The gate requires high fidelity $|A\rangle$ states, and the total number of corresponding distillations combined with their resource requirements is so high that it almost monopolizes the cost of error correcting an arbitrary quantum computation.

Instead of a conclusion, our quantum computing engineer should not be afraid of defects and be motivated to investigate ways to optimize distillations and their placement in assemblies (see Figure 5). The engineer can learn more about this topic by using complete introductions to the surface code [1, 5] and the description of how defect assemblies are generated [3].

References

[1] Fowler, A. G., Mariantoni, M., Martinis, J. M., and Cleland, A. N. Surface codes: towards practical large-scale quantum computation. *Physical Review A 86*, 3 (2012), 032324.

[2] Bravyi, S., and Kitaev, A. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Physical Review A 71*, 2 (2005), 022316.

[3] Paler, A., Devitt, S. J., and Fowler, A. G. Synthesis of arbitrary quantum circuits to topological assembly. *Scientific Reports* 6, 30600 (2016).

[4] Paler, A., Polian, I., Nemoto, K., and Devitt, S. J. A compiler for fault-tolerant high level quantum circuits. 2015. arXiv preprint arXiv:1509.02004.

[5] Raussendorf, R. and Harrington, J. Fault-tolerant quantum computation with high threshold in two dimensions. *Physical Review Letters* 98, 19 (2007), 190504.

Biographies

Dr. Alexandru Paler is a postdoc at the Johannes Kepler University Linz. His research focuses on algorithms for assembling and executing error-corrected quantum circuits.

Dr. Austin Fowler works for Google as a quantum hardware engineer. He specializes in surface code quantum error correction and is currently engaged in writing software capable of tracking errors in such codes sufficiently quickly to keep pace with an array of superconducting qubits.

Dr. Robert Wille is a full professor at the Johannes Kepler University Linz and head of the Department for Integrated Circuit and System Design. Before that, he worked as Ph.D. student and postdoc at the University of Bremen as well as the German Research Center for Artificial Intelligence (DFKI) and was a visiting professor at the University of Potsdam and the Technical University Dresden. His expertise is in the development of design technologies for both conventional and emerging circuit technologies.

Establishing Quantum Advantage

What are quantum computers good for? This essay reviews the progress toward proving a quantum advantage over classical computing.

By Adam Bouland

DOI: 10.1145/2983543

Quantum computing is a method of manipulating tiny particles, such as individual photons or electrons, to process information. These particles obey the laws of quantum mechanics, which are fundamentally different from the laws of physics observed in day-to-day life. For example, particles can hold different positions and momenta simultaneously—a property known as quantum superposition. By harnessing the strange laws of quantum mechanics, quantum computing promises to provide an exponential advantage over classical computation. In other words, quantum computers could solve some problems in time, which scales polynomially in the input size, whereas a classical computer would require time that scales exponentially in the input size. This "quantum leap" in computational power is widely proclaimed as revolutionary in popular media. In 2013 Google uploaded a video to YouTube of a scientist who claimed in addition to solving more mundane optimization problems, quantum computers might even help us determine whether or not extraterrestrials exist [1]. In the face of such speculative pronouncements, it is natural to ask exactly what problems quantum computers (once they're constructed) will be able to solve considerably faster than classical computers, such as laptops and smartphones. In other words, what are the "killer apps" of quantum computing? And is there evidence that proves classical computers (which we are far better at constructing) cannot perform these tasks efficiently? In this article we review the problems that show a quantum advantage, discuss the state of mathematically proving this advantage, and describe recent research on sampling tasks, which might provide the first experimental demonstration of quantum advantage in the near future.

Image by Agsandrew/Shutterstock.com

TASKS WITH QUANTUM ADVANTAGE

There are a number of problems where quantum computing may potentially have an exponential advantage over classical computation. For example, quantum computers would have the incredible power to break most of the modern cryptography. In 1994, Peter Shor demonstrated quantum computers could factor an n-bit number in time that scales polynomially in n [2]. In contrast, the best-known classical algorithm takes time that scales exponentially in n. Many believe more efficient algorithms do not exist, and this belief is the basis of much of modern cryptography. In fact, quantum computers could efficiently break a number of public-key cryptosystems, including RSA and Diffie-Hellman, which are used in electronic commerce and banking. So an adversary with a quan-

tum computer could break into customers' bank accounts, read encrypted emails, and interfere with online shopping. Fortunately, there are some alternative cryptographic protocols, such as lattice-based cryptography, which may be immune to quantum attacks. Nevertheless, the development of quantum computers will radically change the nature of cryptography.

Although the cryptographic applications of quantum computers are well advertised in the popular press, a potential killer app of quantum computing, and one that is rarely mentioned, is the task of simulating quantum mechanics. This was part of the original motivation for building quantum computers. In 1982, Feynman observed to simulate a quantum system of n particles, one needed to keep track of roughly 2^n parameters. Therefore, he reasoned it might be difficult to simulate large quantum systems on classical computers. This has turned out to be true in practice as simulating complex quantum mechanical processes, such as protein folding, are extremely difficult. In practice, researchers use various simplifications or approximations to quantum mechanics, such as density functional theory, to estimate their quantities of interest. However, these heuristic approaches work in limited cases. If quantum computers are built, this will allow us to simulate many quantum processes to high precision. These simulations would have many applications in pharmaceuticals, materials science, and quantum chemistry.

Additionally, we believe quantum computers can provide an exponential advantage for more obscure problems, like computing certain topological invariants of knots. Strangely these problems have been shown to be complete for quantum computing. This means that any quantum computation can be recast as a problem of computing topological knot invariants. However, it is unclear what the application of such results would be outside of knot theory.

Interestingly, we do not have evidence that quantum computers will have an advantage solving *NP*-hard problems. An *NP*-hard problem is a problem for which one can efficiently verify the solution, but cannot necessarily find the answer efficiently due to the fact that there are exponentially many possible solutions. For example, finding the most efficient route to visit n cities is an *NP*-complete problem. There are n factorial orderings of cities one could visit, and it is difficult to find what ordering results in the shortest travel time. Such problems commonly arise in optimization, where it is extremely difficult to find the best solution to a complex optimization problem. Although D-Wave, a commercial quantum computing company, is working to construct quantum computers to solve such optimization problems, it is unclear if quantum computers will have an advantage at this task. There is evidence that quantum computers cannot efficiently solve *NP*-hard optimization problems on worst-case inputs. In particular, we know quantum computers cannot efficiently brute-force search over an exponentially large set of all possible answers to find a solution. In other words, quantum computers aren't any better at finding "a needle in a haystack" than a normal computer, up to polynomial factors. Therefore, it is widely expected that quantum computers will not be able to solve *NP*-hard problems exponentially faster than classical computers on worst-case instances. It remains open whether quantum computers can provide an advantage on natural or average-case inputs of *NP*-hard problems, as they are often much easier than worst-case inputs. This is also the subject of a current experimental investigation by D-Wave and the research community.

LOCKING IN THE ADVANTAGE

So far we know quantum computers can solve factoring, simulate quantum mechanics, and compute knot invariants exponentially faster than the best-known classical algorithm. But how do we know these problems can't be solved with your laptop? What if a faster classical algorithm exists, but algorithms designers simply haven't thought of it yet? Can we rule out this possibility?

Unfortunately, the short answer to this question is "no." Ruling out the existence of efficient classical algorithms for such problems is an extremely difficult mathematical problem. In essence, this is difficult because infinitely many possible algorithms exist, essentially

one for each computer program one could write. While it is obvious most computer programs one could write will not solve factoring, it is very difficult to argue none of those algorithms would work, or none of those that work would be efficient. It is impossible to go through them individually and figure out why they do not work, as doing so would take an infinitely long time.

In fact, if it can be proven that quantum computers have an exponential advantage for any of these tasks, it would answer a major open problem in computational complexity theory. An established result is a classical computer using polynomial memory and exponential time can also solve any problem that can be solved by a quantum computer. Such an algorithm is called a PSPACE (polynomial space) algorithm, because it uses a reasonable amount of memory but an unreasonably long amount of processing time. Therefore, to prove quantum computers have an exponential advantage over classical computers, it would require proving a more efficient algorithm with a faster runtime could not solve any PSPACE problem. This is known as the *P* vs. PSPACE problem. Interestingly, this is an open problem, which is closely related to the *P* vs. *NP* problem. The latter problem carries a $1M prize from the Clay Mathematics Institute for its solution.

Mathematicians have a long way to go before they can solve either of these problems. In particular, we know several proof techniques, which are useful in other areas of complexity theory, will not be able to resolve the *P* vs. *NP* problem, the *P* vs. PSPACE problem, or the problem of proving a quantum advantage. For instance, imagine a quantum computer or a classical computer is given access to a subroutine, which computes some arbitrary function *f* in constant time. Each computer is provided access to this function, as an API, but has no control over how it is being solved in the back end. The strange thing is, there exist some functions *f* that render quantum computers and classical computers equally powerful in this setup. So any argument made that is independent of the APIs available cannot prove quantum computers are more powerful than classical computers. This is known as the "relativization" barrier in complexity theory. Unfortunately, most arguments we know to prove separations in power are invariant under giving each computer access to the same APIs, so any proof of that form will not be able to resolve this problem.

By harnessing the strange laws of quantum mechanics, quantum computing promises to provide an advantage over classical computation.

Since common proof techniques do not work for this problem, mathematicians do not know where to begin to approach these problems. We do know how to prove some similar, but much weaker, statements. For example, we can sometimes show algorithms with power *A* cannot be simulated by algorithms with power *B*, but only when the power of *A* is drastically more powerful than *B*. Even these results are considered breakthroughs in the field. As a result, it seems there is no hope of definitively ruling out the possibility of classical algorithms for our applications of quantum computing—at least in the near future. Furthermore, it is not known solving the above problems in classical polynomial time would have any unexpected consequences in structural complexity theory, so we cannot base our belief in quantum advantage for these problems on other assumptions in complexity theory. It appears our quest to mathematically prove quantum advantage is at a dead end.

POTENTIAL NEXT STEPS

In the face of these obstacles, there are several approaches one can take toward establishing a quantum advantage. One approach is to dismiss the concerns of the previous section as merely theoretical. Rather than mathematically proving quantum advantage for these problems, one can simply try to build a quantum computer to solve very large instances of these hard problems. The hope is one will empirically, rather than theoretically, demonstrate a quantum advantage. For example, if an integer is factored with a quantum computer, which is larger than what any classical computer has ever factored (e.g. the RSA 2048-digit challenge key), this would be an impressive experimental evidence of quantum advantage.

While this is a long-term goal of the experimental quantum computing community, we cannot yet build quantum computers that can factor large numbers, despite much experimental progress. So far we can only build prototype quantum devices, and the largest number ever factored by these prototypes using Shor's algorithm is currently 21. So it seems an empirical demonstration of quantum advantage by factoring large integers is beyond our present experimental abilities. A natural practical question arises: What are these prototype quantum computers good for? Can we find some problems that can be solved with a small quantum computer, which still give an advantage over classical computation? Perhaps quantum advantage could then be empirically tested using our prototype quantum devices, before we are able to build bigger and better quantum computers capable of factoring.

Another, seemingly opposite approach is to dig deeper into the realm of theory, and find different problems, which quantum computers have an advantage solving. The hope is that one may be able to base one's belief of quantum supremacy on different complexity-theoretic assumptions other than the hardness of factoring or simulating quantum mechanics. Perhaps using this approach, one could provide new theoretical evidence for quantum advantage.

A NEW APPROACH: SHOWING QUANTUM ADVANTAGE FOR SAMPLING

Amazingly the two approaches mentioned have recently converged. It turns out by studying a different type of computational task, known as a sampling problem, one can establish quantum advantage on different theo-

retical grounds. Furthermore, these sampling problems might be easier to implement practically than factoring, and might lend themselves to experimental implementation before we have better quantum devices that can factor. In short, one might be able to provide both stronger experimental evidence and further theoretical evidence of a quantum advantage.

Sampling problems are entirely different from how we normally think of computation. Instead of asking for a particular output for every input, a sampling task asks you to sample from a particular probability distribution on outputs for every input. For example, a sampling task might be: Given a number x in binary, sample a random binary string that contains x as a substring. The basic idea is to consider what kind of probability distributions (rather than deterministic functions) can be sampled from with quantum computers. This is an entirely different notion of a computational task.

Recently, research has shown quantum computers can perform sampling tasks that classical computers cannot do. One can show these are impossible for classical computers, assuming a widely held conjecture in complexity theory known as the "non-collapse of the polynomial hierarchy" (a generalization of the belief that $P \neq NP$). As a bonus, often these sampling tasks can be performed with weaker quantum computers that do not necessarily have the ability to factor, so they are particularly well suited to be performed by the prototype quantum computers we are able to build experimentally thus far. For example, the boson-sampling model of Aaronson and Arkhipov [3] can be implemented using linear optical quantum computers without interactions between the photons. In contrast, factoring with optical quantum computers requires nonlinear interactions between the photons, which are difficult to implement. Similarly, the "Temporally Unstructured Quantum Computing" model of Bremner, Jozsa, and Shepherd [4] can be implemented using operations that commute with one another. These may be easier to implement in practice on superconducting qubit architectures. Both these models can perform sampling tasks that are difficult for classical computers, and yet may be easier to implement than factoring. Even performing these tasks on small prototype quantum computers, with around 20-40 quantum bits, would be beyond the capabilities of a laptop. In contrast, factoring 40-bit numbers can be performed on a laptop.

It is widely expected quantum computers will not be able to solve *NP*-hard problems exponentially faster than classical computers.

Admittedly, these sampling tasks are less useful than factoring or simulating quantum mechanics. They sample from probability distributions, which do not help to solve any hard problems. However, their principal utility lies in demonstrating quantum advantage from prototype quantum computers, which will be constructed in the near future. Whatever these tasks are, they capture something that cannot be done with a classical computer.

There are some theoretical difficulties with this approach as well. To achieve the hardness of simulation results mentioned earlier, one has to consider the probability distributions output by perfect, noiseless quantum devices. But in practice, quantum computers are inherently noisy, due to the presence of unwanted interactions between the quantum computer and its environment. By using clever quantum error correcting codes, one can decrease the amount of noise such that it doesn't affect the computation much. This residual noise is irrelevant for solving problems like factoring, but is much more troublesome for these sampling tasks. In fact, it is troublesome enough to foil the hardness results for the sampling tasks mentioned previously. To overcome this obstacle, a number of researchers have sought to prove hardness of simulation under more realistic error models. So far researchers can only prove such results under unproven mathematical conjectures. It seems very difficult to remove these sorts of assumptions from the proofs. But if one proves the conjectures, it would show that even noisy quantum devices have supremacy over classical devices for sampling tasks (assuming the widely-held generalization of $P \neq NP$).

CONCLUSION

Quantum computers are capable of performing a number of tasks exponentially faster than classical computation devices. In the long term, we expect to find a quantum advantage for factoring and simulating quantum systems, which will fundamentally change cryptography, biotechnology, quantum chemistry and materials science. However, mathematically proving quantum advantage is surprisingly difficult, and empirically demonstrating a quantum advantage for factoring is beyond our current experimental capabilities. In the near future, we may have access to prototype quantum computers acting on a few tens of quantum bits. Rather than using these computers to factor integers, it is likely we will use them first to perform sampling tasks that show a quantum advantage over classical devices. These sampling tasks will be a stepping-stone on our way to unlocking the full power of quantum computation.

References

[1] Google and NASA's Quantum Artificial Intelligence Lab. YouTube. October 11, 2013 https://www.youtube.com/watch?v=CMdHDHEuOUE, Time 4:57

[2] Shor, P. W. Algorithms for quantum computation: discrete logarithms and factoring Proc. *IEEE FOCS'94*, (1994),124-134.

[3] Aaronson,S., and Arkhipov, A. The computational complexity of linear optics. *Theory of Computing* 9, 4 (2013), 143–252.

[4] Bremner, M. J., Jozsa, R., and Shepherd, D.J. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 467, (2010), 459.

Biography

Adam Bouland is a Ph.D. student at MIT studying theoretical computer science. Prior to coming to MIT, he completed his undergraduate at Yale and his master's at the University of Cambridge under a Marshall Scholarship. His research focuses on classifying the computational power of weak models of quantum computing as well as finding connections between quantum computing theory and physics.

1528-4972/16/09 $15.00

Programming Quantum Computers Using 3-D Puzzles, Coffee Cups, and Doughnuts

Programming a quantum computer is a task as baffling as quantum mechanics itself. But it now looks like a simple 3-D puzzle may hold the solution.

By Simon J. Devitt

DOI: 10.1145/2983545

Andrew Steane, one of the pioneers of quantum computing, once quipped: "A quantum computer is an error correction machine—computation is just a byproduct." Steane provides an extremely apt description of how any large-scale active quantum technology will ultimately behave. Quantum information processing suffers from two disadvantages: Controllable quantum bits, or qubits, are extremely susceptible to noise from bad control or the external environment; and quantum algorithms are, by nature, exceedingly sensitive to errors. Even a single error during the execution of an algorithm can lead to, essentially, random output.

Hence, quantum error correction (QEC) was quickly recognized as a necessity for any commercially viable computational or communications protocol, and the theoretical development of error correction techniques is as old as the first architectural models for quantum computers. It is thanks to the work of researchers such as Peter Shor, Andrew Steane, Alexi Kitaev, and Robert Calderbank that QEC was developed in the mid 1990s. When combined with the principle of fault-tolerant quantum computation, QEC leads to arguably the most important theoretical result in quantum computing: the threshold theorem.

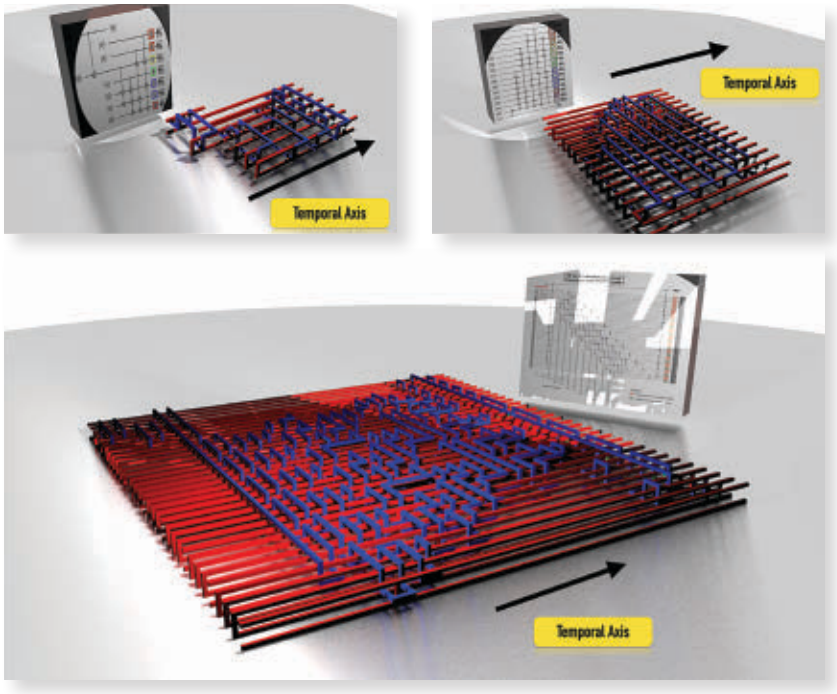
A quantum computation of arbitrary size can be completed successfully with faulty qubits, with a polylogarithmic resource overhead, provided that the physical error rate associated with each qubit and logic gate is below a maximum value, dubbed the fault-tolerant threshold.

What this theorem is basically saying is provided the error experienced by each qubit is below a certain value (the threshold), error correction will correct more errors than it introduces, and a computation, no matter how large, will always be error free by introducing extra qubits.

The value of the fault-tolerant threshold is determined by many factors: the QEC code utilized, the way error correction codes are constructed, and any physical restrictions of the quantum hardware, such as if qubits can be coupled together arbitrarily or are interactions restricted to a fixed geometry. Initial estimates were very unfavorable,

Figure 1. Canonical topological quantum circuits.

In each figure we illustrate a quantum circuit (written in the standard pictorial form) and the corresponding un-optimized topological quantum circuit. Each circuit can be measured in terms of volume. The temporal axis is defined as the temporal evolution of this structure as the computation proceeds.



with thresholds for the Steane code, and other models, of the order of 0.01 percent. However, this has improved dramatically with the development of topological models of QEC [1]. These exhibit fault-tolerant thresholds approaching 1 percent for models such as the surface code and the Raussendorf code. These codes are also much more amenable to physical implementation, as they are defined on a two-dimensional (surface code) or three-dimensional (Raussendorf code) array of nearest-neighbor interacting qubits.

The high fault-tolerant threshold, the nearest-neighbor nature of these topological codes, and the way in which quantum algorithms are implemented have resulted in them becoming the preferred technique for large-scale quantum computing architectures. Essentially, all major physical system are now targeting either the surface code or the Raussendorf code for their architectures, and physical systems such as ion traps and superconducting qubits are now demonstrating gate and qubit error rates either at, or below, the fault-tolerant threshold. It is becoming increasingly probable that a functional, commercial quantum computing system will be build using topological QEC as the fundamental computational model.

In the current issue of *XRDS*, Paler et al. have compiled a review that details how both computation and error correction is performed in topological QEC models. This article continues that discussion, and examines both the structure of a topological quantum circuit and how these circuits will ultimately be optimized and implemented on a real world quantum computer.

TOPOLOGY: COFFEE CUPS AND DOUGHNUTS

Topology, unsurprisingly, plays a crucial role in the function and operation of topological quantum error correction. As with, essentially, all QEC codes that are considered implementable on large-scale hardware, topological quantum codes are defined in terms of stabilizer operators. A quantum state $|\psi\rangle$ is stabilized by an operator K, such that $K|\psi\rangle = |\psi\rangle$. A topological quantum code is defined by a set of these operators, which are defined locally. That is, they are defined over a small group of qubits that are near each other. However, the encoded state defined by these operators has certain global properties. Logical operations, those that define the encoded qubit state, are defined with respect to the entire state—they cannot be defined locally. This is the essential nature of a topological code. Individual stabilizers used to perform error correction are defined locally, while logical information is defined globally.

As summarized by Paler et al. a two-dimensional lattice of qubits (for the surface code), or a three-dimensional lattice of qubits (for the Raussendorf code) defines a unique quantum state. The eigenvalues of each of the stabilizers associated with the lattices are measured in order to detect and correct for quantum errors that can occur due to imperfect physical qubits and gates. Information is encoded into this lattice through the creation of holes, or defects. These are regions of the lattice that have been deactivated by having qubits removed. By removing qubits, or deactivating certain parts of the lattice, degrees of freedom are introduced into the quantum state that can be used to store and manipulate information, which is protected from errors due to the properties of the remaining lattice—called the bulk.

Interactions (quantum gates) in this model are enacted through an operation called "braiding." Braiding is the perturbation of defects such that they "move" through the lattice as it evolves in time, and wrap around each other like a tangled ball of string. An example of a large topological circuit that enacts a set of logical gates on encoded qubits, in a fault-tolerant way, is illustrated in Figure 1. The spatial cross section is illustrated, as well as the temporal axis. The spatial cross section defines the number of qubits used in the surface code while the temporal axis defines how defects are created and manipulated over time. In the Raussendorf model, all three dimensions of the lattice consist of

physical qubits, which are sequentially measured along the temporal axis. Measurements are used to define and manipulate the defects through teleportation along this temporal axis of the Raussendorf code.

These structures are topological in nature and, hence, standard definitions of topology apply. The nature of a topological space is that it is preserved through operations known as continuous deformation. Continuous deformation is where a structure is stretched or bent without being cut or glued together at any point. The quintessential example of this is the topological equivalence between a coffee cup and a doughnut shown in Figure 2a. Simply by stretching and bending, the coffee cup can be converted to a doughnut, and vice versa. As each structure has only a single hole, they are topologically equivalent.

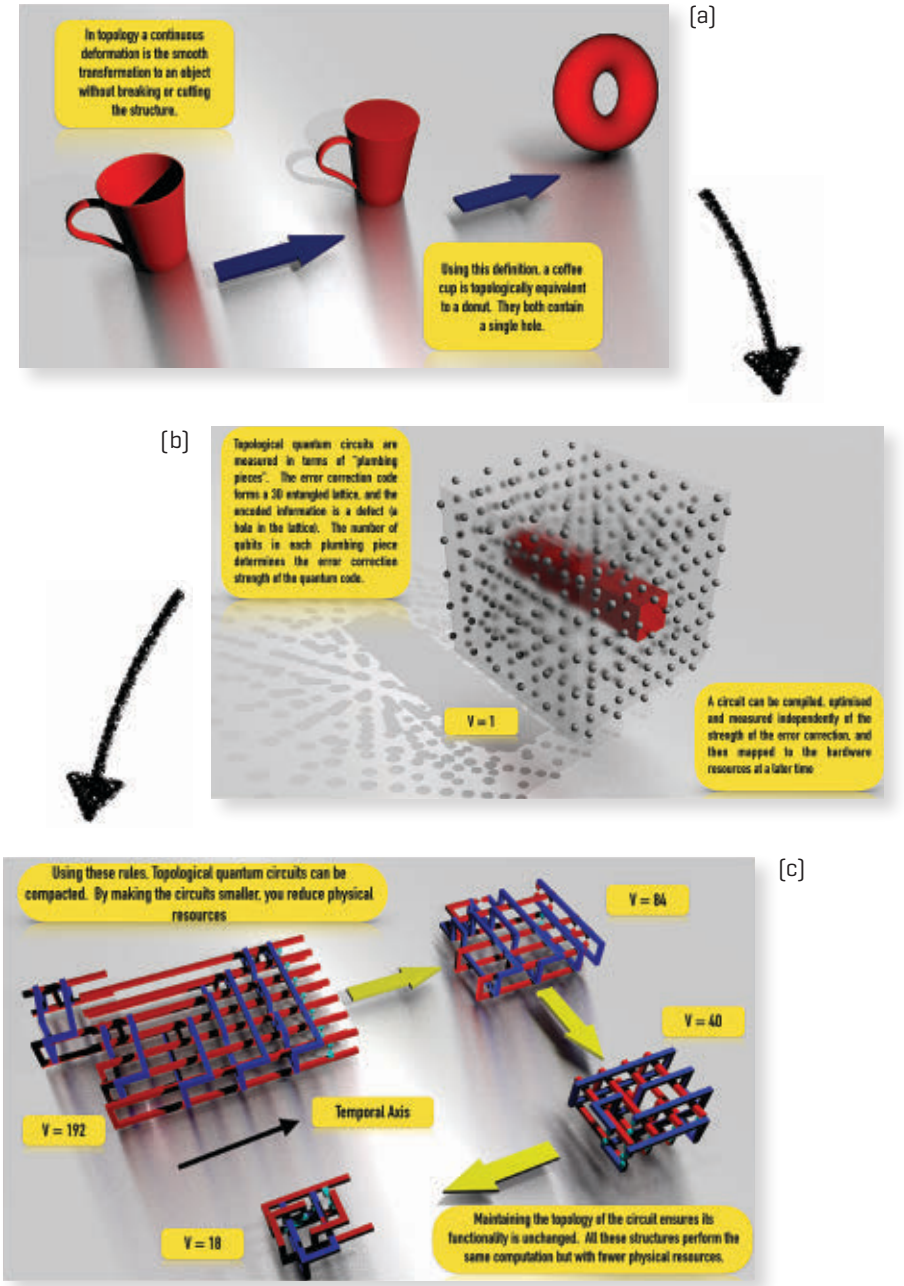
Therefore, a quantum program is, literally, defined and described by a puzzle. This puzzle can be shaped, stretched, and molded to change the physical resources needed by a quantum program without changing the program itself.

MEASURING AND BENCHMARKING QUANTUM CIRCUITS

In order to derive relevant metrics when constructing, compiling, and optimizing topologically error-corrected circuits we need to understand how a circuit relates to the number of qubits and the physical computational time when they are implemented [2]. Regardless of whether we are talking about the surface code or the Raussendorf lattice, the relationship between a topological circuit and physical resources is identical. The fundamental unit of measure is illustrated in Figure 2b. A "plumbing piece" of a topological quantum circuit is a three-dimensional cubical volume that has an edge length related to the desired strength of the underlying quantum code. For a distance d code, sufficient to correct up to $t = (d–1)/2$ errors, this plumbing piece has an edge length containing $5d/4$ plaquette cells for the surface code, or $5d/4$ cells in the Raussendorf lattice. At the center of this plumbing piece is the defect, which has a circumference of d plaquettes. Figure 2b illustrates an example for $d = 4$. The plumbing piece gives a scale independent factor to allow us to measure topological quantum circuits without having to specify the strength of the underlying error correction. Using topological circuit volumes in terms of plumbing pieces allows us to directly calculate the total number of qubits, and the computational time. For the surface code, the plumbing piece requires a total of $Q = 25d^2/4 + 5d + 1$ qubits and

Figure 2. How to optimize topological quantum circuits.

Figure a represents the topological equivalency of a coffee cup and a doughnut. Figure b illustrates a "plumbing piece," the basic unit of measure for a topological quantum circuit. Increasing the error correction code required more physical qubits for each pluming piece. Figure c shows how topological deformation can be used to reduce the volume and, hence, physical resources, without changing its information processing properties.



$T = 5d/4$ steps. Each step is defined as a syndrome extraction circuit for both bit-errors and phase-errors. For the Raussendorf lattice, a plumbing piece requires a total of $Q = 6d^3 + 9d^2 + 3d$ qubits. For a larger topological circuit, we can use their volume to first calculate the required strength of error correction d to ensure no logical errors occur during implementation, and then calculate the total number of resources needed by converting the volume to physical qubit numbers and computational time.

This method of designing quantum programs is very useful, as we do not need to redesign anything about the actual quantum hardware when we change the quantum program. We just need to make sure we have enough qubits to do the job.

CONSTRUCTING AND COMPILING INITIAL TOPOLOGICAL CIRCUITS

Before a given computation can be suitably optimized in the topological formalism, quantum circuits need to be compiled and constructed from the original algorithmic specification. Figure 3 illustrates the broad structure of the compilation stack needed for an arbitrary algorithm. The stack is partitioned into several stages:

1. **Algorithm to circuit.** A quantum circuit, consisting of single, two, and three qubit gate primitives, is derived from the abstract algorithm. This circuit can be optimized for depth and number of qubits.
2. **Circuit to fault-tolerant primitives.** The abstract circuit is further decomposed into gate sets that have well defined fault-tolerant implementations in the topological code. Again, optimizations can occur at this level
3. **Fault-tolerant circuit to ICM form.** The circuit consisting of fault-tolerant gate primitives is then converted to a form called initialization, controlled-not, measurement (ICM). This allows us to build in appropriate auxiliary protocols needed before explicitly converting them to a topological implementation.
4. **Canonical topological form.** Once written in ICM form, the circuit can be converted to an un-optimized canonical form in the topological model, prior to further resource optimization.

The conversion of a higher level circuit to a canonical topological form is a complicated but well-defined process, and researchers have designed several software packages to perform this task.

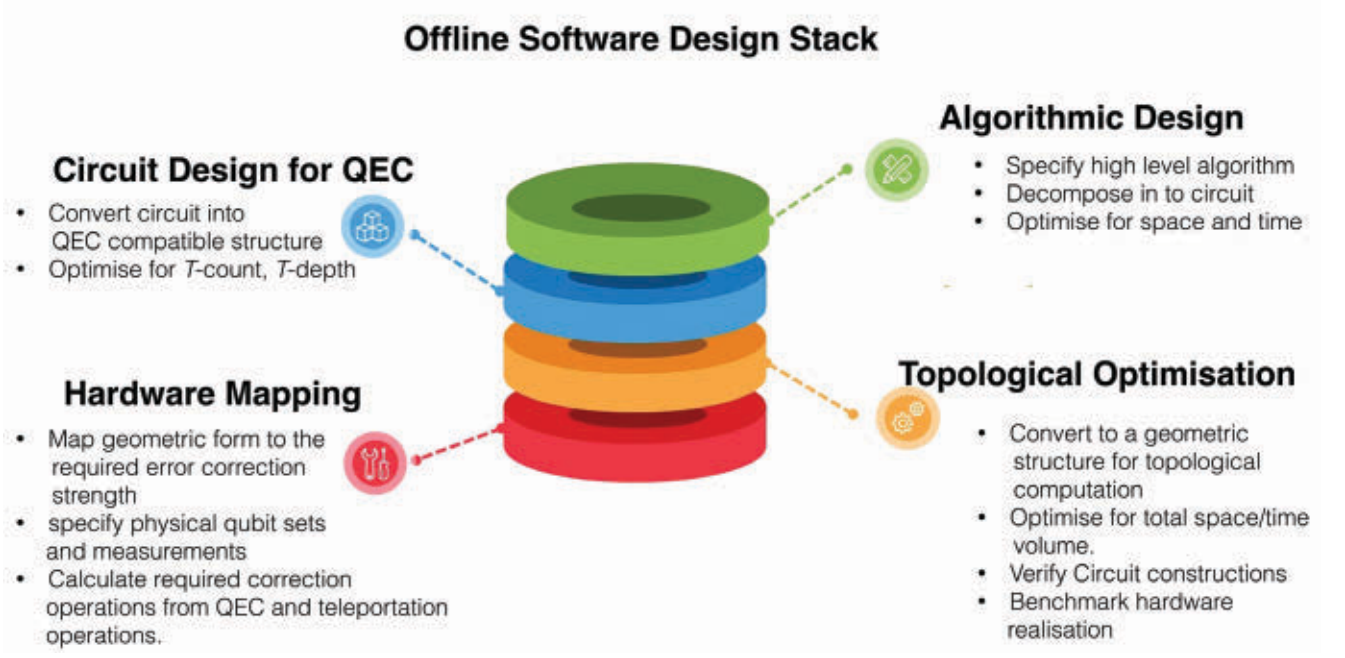
In Figure 1 we illustrate several examples of a canonical topological form and the quantum circuits they are derived from. Each of the circuits shown are known as "magic state distillation circuits" and are used to enact certain gates that require high-fidelity ancillary states. Each of these circuits have a corresponding volume and, hence, can be used to estimate physical resources.

TOPOLOGICAL OPTIMIZATION

The next crucial step in the design stack for error-corrected quantum circuits is topological optimization. This also happens to be the most underdeveloped area of the stack. Almost all other elements have been completely understood, or are being

Figure 3. Offline compilation and optimization stack.

The conversion of an abstract, high-level quantum algorithm to an un-optimized topological form. What is addressed in this review is the third level, which corresponds to topologically compacting a given circuit specification prior to hardware implementation.



heavily researched, including efficient methods for circuit construction and optimization and fault-tolerance. Although it has received little attention, there are strong reasons to believe topological optimization may result in some of the largest resource savings if implemented well.

The basic principal is illustrated in Figure 2c. The canonical circuit begins with a volume of $V = 192$ plumbing pieces. In the same way as a coffee cup can be deformed topologically into a doughnut, we can slowly compact the physical three-dimensional volume of the circuit without altering its computational function. There are some additional rules not related to continuous deformation, and unique to the computational model (such as "bridging") that can be used to reduce the physical volume of a topological circuit significantly. After many steps (not illustrated), the final volume of the topological circuit is reduced to $V = 18$, over an order of magnitude smaller than the original canonical form. This amount of optimization is indeed significant. For a surface code quantum computer, the number of qubits required for implementation can be reduced by orders of magnitude by simply compressing these structures.

However, two theoretical questions still remain unanswered.

The first is to provide a lower bound, or exact definition for optimality, for a topological circuit. While we can compress, we do not have a condition for optimality given the original circuit specification. The second question concerns the classical complexity of the algorithm required to find this optimal solution. While this problem does appear closely related to the three-dimensional bin packing problem, which is known to fall into the complexity class of *NP*-hard, there are small differences in the topological QEC model that may imply that these two problems do not directly map on to each other. It is still possible optimization of topological quantum circuits may be provably and classically efficient to calculate.

Circuits that have been currency compacted have been done so manually. This is obviously not a viable approach for large-scale implementations of error-corrected quantum algorithms. There have been very small steps to try and build automated topological optimization packages, but these have, so far, only illustrated the potential difficulties in creating the required software. Being able to optimize even moderately large quantum circuits will not be possible without automated software, and it appears as though techniques in machine learning and artificial Intelligence may be required to provide resource efficient solutions.

Figure 4. Current assets for the prototype client of meQuanics.

Shown here is a current screenshot from the meQuanics client [www.mequanics.com.au], and some digital assets that will be used for the final game client.

THE QUANTUM COMPUTER GAME

The approach we recently took to address this problem was inspired by projects in the biological sciences that attempt to solve scientifically useful, but difficult, problems by utilizing the computing capacity of the general public. This technique, sometimes referred to as "citizen science," was pioneered by projects such as FoldIt (which aimed to find the three-dimensional structure of biological proteins given their constituent sequence of amino acids), and Eyewire (designed to map neural connections in the retina), and has achieved significant success.

Given the relatively simple 3-D puzzle structure of topological quantum circuits, and the simple success metric of minimal physical volume, we have tried the same approach. An initial prototype of a platform we have dubbed meQuanics (www.mequanics.com.au), designed to convert the to-

pological optimization problem into a simple 3-D puzzle game, has been released online. Designed for touch-based platforms such as smartphones and tablet devices, meQuanics creates an online social media environment where the general public can compete and collaborate to find small volume solutions to various quantum sub-circuits, which are critical for large-scale quantum computation.

While it is conceivable users can derive compact solutions that are significantly smaller than solutions we currently have, the primary goal is not the solutions but rather the process individual players use to generate these solutions. It is well known within the machine learning and AI community that the success of these techniques requires a database of information that the machine system can use to learn. While for many problems there is an existing database of material that can be utilized (such as the AlphaGo platform of the DeepMind project at Google), for this particular problem there are, essentially, no previous examples that we can use to train an appropriate automated program.

While the prototyping stage has demonstrated proof-of-principle client, there is still significant development required. The most important goal is to create a game that is competitive in the larger marketplace of mobile and touch-based games. While the novelty of a game that is very closely related to quantum computing development may induce a large number of users to try it out, long-term retention of gamers is required to generate the necessary data sets for the project to be successful.

Gameplay and social interaction in meQuanics is now a major focus of development. The basic narrative is there is an interstellar race, where each ship is powered by a quantum computer. By minimizing the volume of puzzles, players increase their speed and ultimately win over other players working on the same problem. The online interaction environment is being designed under the assumption that each client continuously updates information to central servers informing us how players are tackling problems. Individual players can take a continuously expanding solution tree, which begins from a specific canonical circuit structure, and either improve on other players solutions or backtrack and proceed down a different pathway that could lead to better solutions. Elements of the social media and gameplay environment are illustrated in Figure 4.

Quantum error correction (QEC) was quickly recognized as a necessity for any commercially viable computational or communications protocol.

THE FUTURE

Future prospects on the software component of large-scale quantum technologies is promising. Not only is there a vast amount of unsolved problems that can be addressed, even by researchers not heavily trained in quantum physics, but the theoretical similarities of essentially all major experimental hardware models implies software solutions are applicable to all systems. This is now evident from the founding of four private startups exclusively focused on the software component of quantum technology: QxBranch (www.qxbranch.com.au), 1Qbit (www.1Qbit.com), QCware (www.qcware.com), and Cambridge Quantum Computing (www.cambridgequantum.com).

While each element of the software compilation stack has been addressed at some level, a functional quantum computer will require a completely integrated set of classical software compilation and optimization packages. The expertise of the classical software engineering community will be vital to this. While physicists may be the experts in building quantum hardware, efficient and reliable software control will probably be developed by those already well versed in the advanced techniques of classical software engineering. The fact that these problems are not intrinsically "quantum" in nature will make it easier for those without explicit training in quantum physics to get involved and make important contributions in this arena.

Quantum information technology is currently experiencing a second renaissance in advancement and investment from both the public and private sectors. As such, there is consensus amongst experts that it is no longer a question of *if* a large-scale quantum computer can be built, but *when*. The quantum revolution has the potential to be as significant as the digital revolution of the 20th century, and there is now a worldwide race to be the first to show a commercial advantage in deploying large-scale computers, communication networks, sensors, and other active quantum technology. We stand at the cusp of an exciting new age in computing, with a significant laundry list of problems to interest pioneers.

ACKNOWLEDGEMENTS

We would like to thank A. Paler and A.G. Fowler for collaborating on research summarized here. meQuanics was developed in collaboration with K. Nemoto, K. Bruegmann, E. Gray, P. Daouadi, M. Everitt, Y. Quemener and F. Schittig. S.J. Devitt acknowledges support from the JSPS grant for challenging exploratory research and the JST ImPact project.

References

[1] Fowler, A., Mariantoni, M., Martinis, J., and Cleland, A. Surface codes: towards practical large-scale quantum computation. *Physical Review Letters* 86, 3 (2012)

[2] Devitt, S., Stephens, A., Munro, W., and Nemoto, K. Requirements for fault-tolerant factoring on an atom-optics quantum computer. *Nature (communications)* 4, 2524 (2013).

Biography

Simon Devitt holds degrees in physics from the University of Melbourne, Australia. His research focuses on large-scale architecture designs for quantum computation and communications systems, and software compilation and optimization for topological quantum computing. He is currently a senior research scientist at the Center for Emergent Matter Sciences at the Japanese research institute, Riken, an advisor to the quantum software company QxBranch and the creator of the "Meet the meQuanics" podcast, which discusses development of quantum technologies.

1528-4972/16/09 $15.00

Black Holes and the Limits of Quantum Information Processing

The densest memories and the fastest processors imaginable on computers located billions of light-years away

By Brian Swingle

DOI: 10.1145/2983549

A black hole is region of space where gravity is so strong not even light can escape. Since nothing moves faster than light, observers who fall into a black hole can never return. As regions of intense gravity, black holes are crucial in physicists' attempts to unify gravity and quantum mechanics. For these reasons, achieving a better understanding of black holes is a major long-term goal of physics research.

For the first time in human history on September 14, 2015, the Laser Interferometer Gravitational-Wave Observatory (LIGO) directly detected gravitational radiation—ripples in the fabric of spacetime—arriving at Earth from a binary black hole merger, an event of cataclysmic proportion occurring a billion light-years away [1]. The discovery reminded us all of the excitement and mystery of black hole physics.

Theoretical calculations of the merger using Einstein's general theory of relativity match the experimental observations quite well. So it is fair to say, the fundamental rules of classical black hole dynamics are reasonably well understood. Nevertheless, this beautiful experiment has given birth to a new era of gravitational wave astronomy, and promises to teach us much about classical black hole physics that we do not yet know.

To make this incredible discovery, LIGO had to measure the relative motion of mirrors four kilometers apart to a precision of one thousandth the width of a proton. At such tiny distances, the world is fundamentally quantum. In fact, in the future, LIGO plans to use a quantum effect known as "squeezing" to make even more precise measurements. Like the advent of gravitational wave astronomy, these achievements can be framed as part of a new era of quantum information pro-

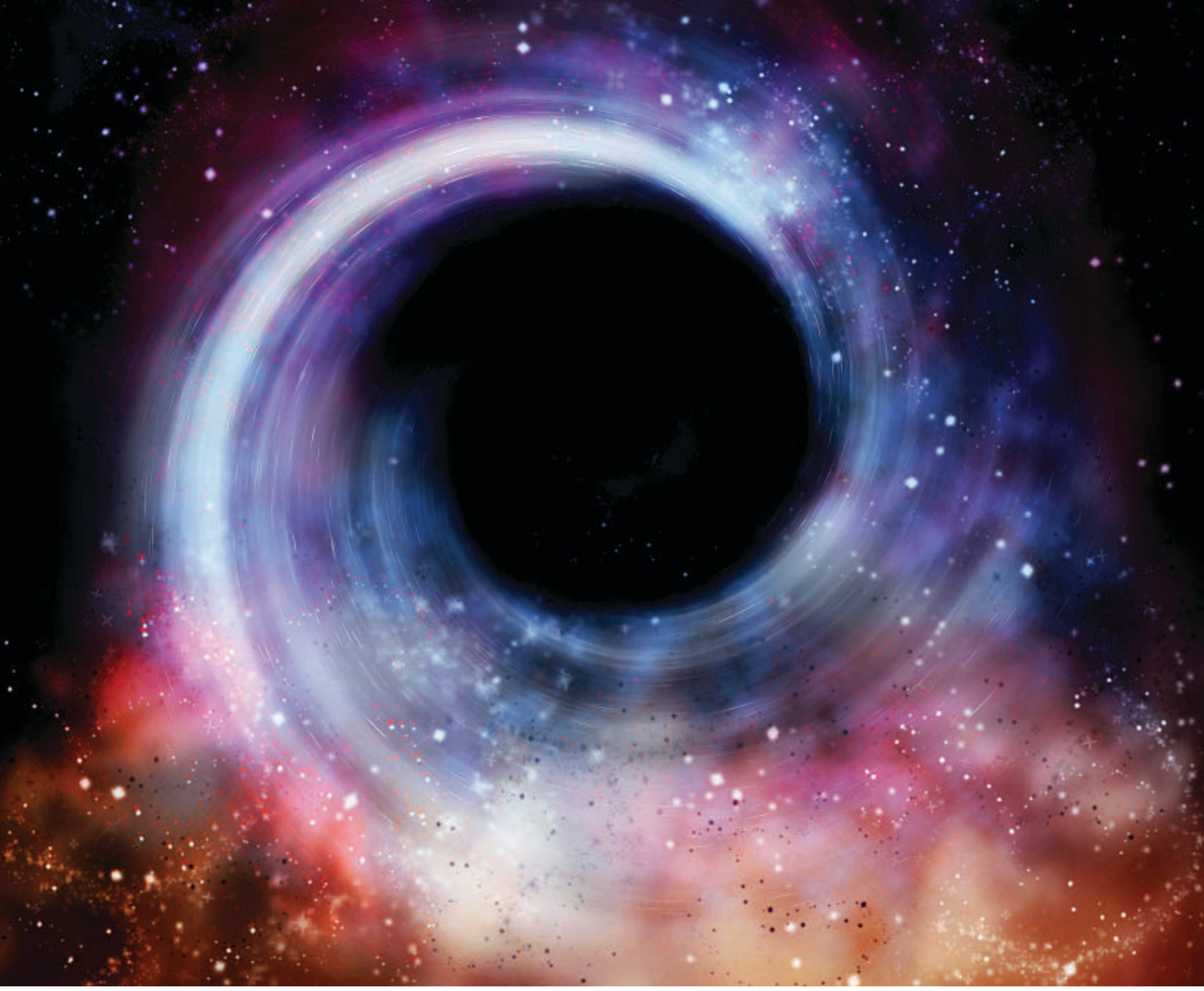


Image by A and Rob/Shutterstock.com

cessing in which we harness quantum physics to define and manipulate information in fundamentally new ways.

A natural question then arises: Is there a quantum description of the black hole itself? This is the much publicized problem of unifying classical gravity and quantum physics, yet the standard answer is no. For the kinds of black holes detected by LIGO, a satisfactory quantum description still eludes us. However, within the theoretical study of quantum gravity, it is useful to consider more general kinds of black holes, starting with a candidate quantum description for a special type of black hole. These special black holes are called black holes in Anti-de Sitter space (AdS).

In the expanding universe in which we live, it is easy to lose energy and information. For example, when electromagnetic waves fly off into the void at the speed of light. By contrast, Anti-de Sitter space acts a like a box, keeping energy and information from running away to infinity. Thus, AdS serves as a particularly simple model in which to study quantum gravity.

To describe black holes in AdS we use powerful mapping known as holographic duality, or AdS/CFT, discovered in the '90s by Juan Maldacena. This mapping makes the following remarkable claim: Certain quantum theories of gravity in asymptotically Anti-de Sitter space are equivalent to certain other quantum theories without gravity in one less spacetime dimension. This mapping is loosely analogous to a soup can—the label on the outside tells you the contents inside.

On the non-gravitational side are certain kinds of quantum field theories called conformal field theories (CFT)—cousins of the standard model of particle physics—which can be regarded as more-or-less conventional quantum systems. On the gravitational side is a theory of fluctuating "quantum geometry" (and, in cases we understand, ultimately a string theory). The duality is called holographic because the gravitational theory emerges from a lower dimensional non-gravitational theory analogous to the way 2-D holograms encode information about a 3-D image.

Returning to the soup can analogy, it is useful to think of non-gravitational theory as living at the boundary of spacetime, while gravitational theory lives in the bulk of the space-

time. The emergent bulk dimension is often called the "holographic direction," or the "radial direction." The conformal field theory is commonly called "the boundary," while gravity theory is called "the bulk." Expressed in this language, the key physical idea is this: If we can understand how to formulate a bulk quantum gravity question using boundary language, then we can use the standard rules of quantum physics on the boundary to answer the bulk question. Thus, we have a candidate theory of quantum gravity provided by this "holographic dictionary" plus the usual rules of quantum physics.

The holographic dictionary is analogous to the following scenario. Suppose someone poses you a hard question in Japanese, but you only speak English. You could learn Japanese to answer this hard question, but the question is quite complex and there aren't any good teachers available. Another way would be to translate the question, word by word, into English using the fixed rules for going between English and Japanese, answer the resulting English question, and then translate it back to Japanese. Quantum gravity theory is like a language we do not yet speak, and the holographic dictionary is a way to translate hard questions in this unknown language into a language we do speak—the language of ordinary quantum physics. In fact, in quantum gravity research we face the even harder problem of finishing the incomplete dictionary as we go.

The question then becomes, what does the holographic dictionary have to say about black hole physics? The physics of quantum information, which we briefly encountered earlier, is actually crucial to understanding black holes in AdS. Indeed, we will see that black holes appear to be extreme quantum information processors, manipulating elementary quantum bits, or "qubits," at the limits imposed by nature. The hope is viewing black holes as extreme information processors may guide us toward an extension of our theory of quantum gravity to more general kinds of black holes.

Let's begin with information storage. If a black hole is to process information, then it must have a memory. Classical black holes "have no hair" and, hence, do not appear capable of storing much information. However, Bekenstein showed black holes have entropy. (Entropy is a measure of the number of internal states of the black hole, that is, the maximum number of bits of information the black hole can store.)

Bekenstein and Hawking showed entropy S is proportional to the area A of the black hole, with the constant of proportionality given by fundamental constants—Planck's constant $\hbar$, the speed of light c, and Newton's constant G:

$$S = \frac{c^3 A}{4G\hbar}$$

This is the first clue to the holographic nature of gravity: Hot ordinary matter has entropy proportional to its volume, but a black hole only has entropy proportional to its area. Hence, it may be described by some effectively lower dimensional information.

Now, one may object that if ordinary matter has entropy proportional to volume, and black holes only have entropy proportional to area, can't we create a sufficiently dense memory made of ordinary matter that has more entropy than the corresponding black hole? This strategy, however, fails. If we tried to make such a high density memory, we would need to stuff so much energy into such a small region that the whole thing would collapse to form a black hole. So, remarkably, not only can black holes store information, they have the densest possible memories.

Information storage is only the start. Black holes also rapidly process their quantum information. If you poke a black hole, it responds to that perturbation by quickly returning to an equilibrium state. Because the black hole has mass (energy) M and entropy S, it is also possible to assign it a temperature $T = \frac{\partial S}{\partial M}$. In terms of this temperature, the time for the black hole to "forget" the perturbation is:

$$t_1 \approx \frac{\hbar}{T}$$

This time, determined just by Planck's constant and the temperature, is a fundamental time scale of any hot quantum system. However, the black hole does not really forget the information in the perturbation. Instead, the information is rapidly spread over the entire black hole so that it becomes inaccessible to any simple measurement—like a memory that stores information in a very complicated and difficult to read form. This process of information spreading is called "scrambling."

Shenker and Stanford, building on work of Sekino and Susskind, used the holographic dictionary to show black holes scramble in time;

$$t_2 = \frac{\hbar}{2\pi T} \log S$$

The scaling of t_2 with the entropy S can be understood as a kind of infection process. Information begins localized in a single bit. Then, roughly every t_1 seconds, each bit carrying some part of the information interacts with another bit. Such an interaction typically spreads the information, first to two bits, then to four, then to eight, and so on. The number of bits carrying the information after n such interactions is of order 2^n. So the total time required to spread information over S bits is approximately t_2. Furthermore, the pre-factor in the scrambling time is precise and saturates a bound proven by Maldacena, Stephen H. Shenker, and Douglas Stanford. Thus, black holes are indeed the fastest scramblers allowed by nature.

So far we know black holes are the densest memories, they forget perturbations rapidly, and ultimately scramble their information as fast as possible. But what happens to the black hole once its information is fully

Black holes are the most extreme quantum information processing devices in nature.

scrambled up? Outside the black hole nothing much appears to happen, but inside it is a different story. Classically, observers who fall into a black hole cannot come back to report what they saw, but the holographic dictionary still asserts that an interior exists, and makes it possible to ask about its holographic interpretation.

Figure 1a shows the spacetime describing a black hole formed from collapsing matter (blue shell) in AdS. This diagram, called a Penrose diagram, is a representation of the causal relationships in the black hole geometry. The rules are that light moves on diagonal lines, and nothing moves faster than light. So, for example, nothing can get out of the black hole because the horizon (the dashed diagonal line in Figure 1a) is effectively moving away at the speed of light.

Long after the information about the black hole's initial condition is fully scrambled, the interior of the black hole continues to grow. There is actually an infinite amount of space in the grey interior region in Figure 1a; the Penrose diagram distorts the infinite space by squishing it all into a finite picture. A simple measure of the black hole interior is the size of the spacetime region shown in Figure 1b (the Wheeler-DeWitt patch) as a function of time (which increases upward in Figure 1). One finds this size increases linearly with time for apparently arbitrarily long times.

What is the holographic dual of this growth? Leonard Susskind, building on work of Thomas Hartman and Maldacena, proposed the growth of the black hole interior is dual to the growth of complexity in the boundary state. Here complexity means "circuit complexity" or the minimum number of elementary quantum operations, generalizations of operations like NOT and XOR, needed to create the state of interest from a standard reference state.

Building on this work, my colleagues and I recently proposed a new entry in the holographic dictionary [2]: complexity equals action;

$$Complexity = \frac{Action}{\pi\hbar}$$

Action refers to the integral of the famous Einstein-Hilbert Lagrangian over the Wheeler-DeWitt patch shown in Figure 1b. Complexity is, again, the circuit complexity of the boundary quantum state, with some choice of the set of elementary operations.

Black holes appear to be extreme quantum information processors, manipulating elementary quantum bits, or "qubits," at the limits imposed by nature.

We showed action grows linearly in time and measures the size of the black hole interior. Furthermore, we were in for a surprise when we computed the rate of increase of action; the time derivative of the action was always twice the mass for black holes of any size and in any dimension:

$$\frac{d\,Action}{dt} = 2M$$

This universality of action growth, along with our proposal that complexity equals action, led us to conjecture black holes generate complexity as fast as possible:

$$\frac{d\,Complexity}{dt} \leq \frac{2M}{\pi\hbar}$$

Our conjecture is similar in spirit to an early proposal due to Seth Lloyd.

Thus, it may be that black holes are the densest memories, the fastest scramblers, and the most rapid complexifiers among all quantum information processing devices in nature. This suggests a new general principle of quantum black hole physics: Black holes are the most extreme quantum information processing devices in nature.

Is there a metatheory that explains why black holes have all these properties? Why are concepts like entropy and complexity, and other ideas from the physics of quantum information at all relevant for understanding the emergence of spacetime and the physics of black holes? In the concluding paragraphs I will present a speculative answer to these questions.

We begin by trying to understand how to constructively generate the holographic dimension, given only boundary degrees of freedom. Some years ago I introduced an approach based on the physics of entanglement in the boundary state [3]. Entanglement is another aspect of quantum information. It is a special kind of quantum correlation that makes it possible to know the state of the whole system perfectly, yet be entirely ignorant of the state of the parts. To connect to our earlier discussion, scrambling can be viewed as a process of entanglement generation.

Figure 1. (a) A spacetime diagram of a black hole formed from collapsing matter. Light moves along 45 degree lines in the diagram. **(b) The region of spacetime whose action is proposed to give the complexity of the quantum state.**

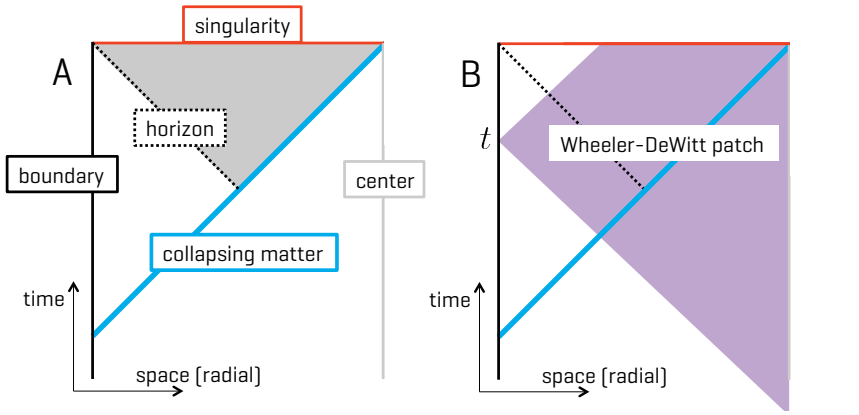


Figure 2. A tensor network schematic of the entanglement structure of the quantum state. In this hierarchical network, each layer corresponds to a particular degree of coarse-graining of the system.

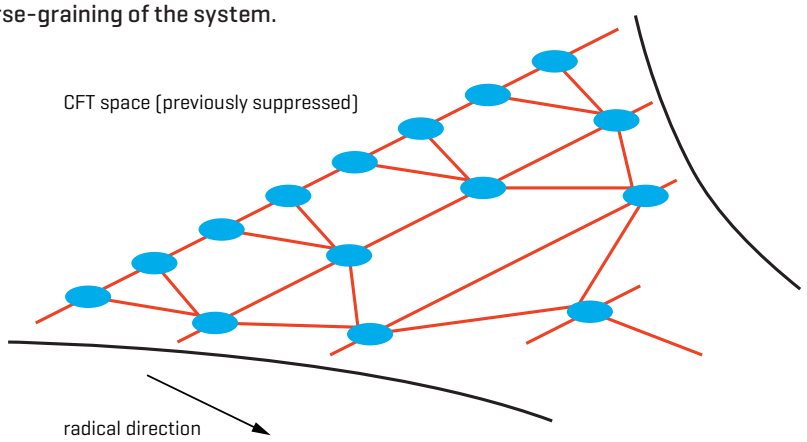
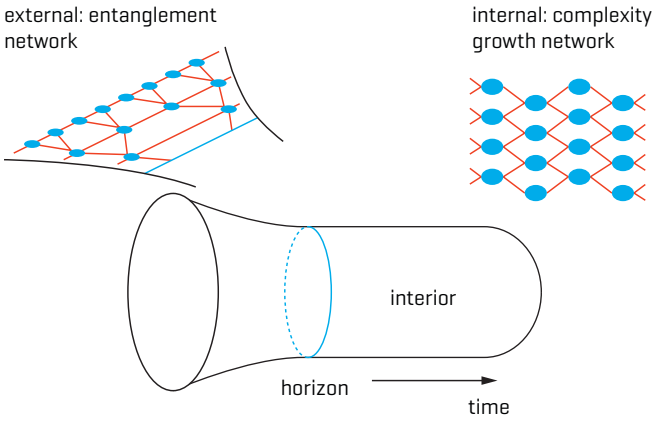


Figure 3. A tensor network schematic of the growing black hole interior. The exterior part of the network encodes entanglement, while the interior part encodes the growth of complexity.



Perhaps, just as there is a dynamical process underlying the growth of the black hole interior, so too is there a dynamical process that builds up the initial state of boundary from simple elements. This process would, in effect, build up the entanglement present in the state scale-by-scale as shown in Figure 2. In this interpretation of the bulk, the radial direction corresponds to an increasingly coarse-grained description of entanglement in the boundary.

In the quantum information community, a picture like Figure 2 is called a tensor network. It is a precise mathematical recipe for building up the quantum state of interest from simple operations, analogous to our discussion of complexity.

Furthermore, it is a general rule of tensor networks that entropy is proportional to area, just like for black holes. I, therefore, proposed entanglement could be viewed as the fabric of spacetime [3]. The tensor network is interpreted as the microstructure of spacetime with the entangling links of the network gluing the points together.

However, entanglement can only be part of the entire story. This is because we stipulated scrambling had already occurred and no more entanglement was being generated, yet the black hole interior grows at late time. Thus we need a new part of the network as shown in Figure 3.

Figure 3 suggests the following general idea: Spacetime is a coarse-grained quantum computational history. More precisely, spacetime emerges as a coarse-grained description of the quantum information processing taking place among microscopic, non-geometric degrees of freedom. Part of the network is associated with entanglement, and part of the network is associated with the growth of complexity, although, there is no sharp distinction. To observers living in the bulk, the geometry is locally the same.

To summarize, one of the main open questions in physics is a theory of quantum gravity. For a special kind of black hole, we have the start of such a theory. This theory involves the new field of quantum information science in an essential way. Weird properties of quantum bits, such as entanglement, seem to play an important role in black hole physics, and the emerging picture is black holes are extreme processors of information. The reason why these concepts are relevant to black hole physics may be that spacetime itself can viewed as recording the history of some quantum computation.

Can these ideas be experimentally tested? Remarkably, the answer is yes. Recall we began with the assertion that theories of gravity could be exactly equivalent to theories without gravity. These non-gravitational theories are of a type that we could, in principle, artificially engineer in the lab. In fact, early versions of these experiments may be just around the corner [4]. Hopefully, the preceding discussion has convinced the reader that when these experiments do happen, we will be ready to test our theories of quantum black holes, and to learn about the fundamental limits on quantum information processing.

References

[1] LIGO Scientific Collaboration and VIRGO Collaboration. Observation of gravitational waves from a binary black hole merger. *Physical Review Letters* 116, 6 (2016).

[2] Brown, A., Roberts, D., Susskind, L., Swingle, B., and Zhao, Y. Holographic complexity equals bulk action? *Physical Review Letters* 116, 19 (2016).

[3] Swingle, B. Entanglement renormalization and holography. *Physical Review D* 86, 6 (2012).

[4] Swingle, B., Bentsen, G., Schleier-Smith, M., and Hayden, P. Measuring the scrambling of quantum information. *arXiv*, February 19, 2016.

Biography

Brian Swingle received his Ph.D. in theoretical physics from MIT in 2011. He was a Simons Fellow at Harvard from 2011 to 2014, and then joined the Stanford Institute for Theoretical Physics in the fall of 2014. His interests span a variety of topics at the intersection of many-body physics, quantum information, and quantum gravity.

1528-4972/16/09 $15.00

Undecidability of the Spectral Gap

What happens to undecidability in the quantum computing paradigm?

By Johannes Bausch

DOI: 10.1145/2983547

There are few papers as influential for a field of research as Richard Feynman's "Quantum Mechanical Computers," published in *Optics News* in 1986 [1] when he was working as a professor of theoretical physics at the California Institute of Technology. As part of an on-going effort to understand the physical limitations inherent to computing, Feynman constructed a physical system that—as a proof-of-concept—could carry out universal classical computation while obeying the laws of quantum mechanics, in particular reversibility. A motivating factor was any irreversible operation must necessarily be accompanied by heat generation due to the intrinsic change of entropy, which is also known as the Landauer limit: An AND-gate destroys one bit of information and thus changes the entropy of the computation state by $ln\ 2$. From a thermodynamics perspective, losing information about a system means losing the ability to extract work from it. Therefore, to operate this gate, a computer running at temperature T has to expend a free energy of $k_B\ T\ ln\ 2$, where k_B denotes Boltzmann's constant (approximately 1.38×10^{-23} Joules/Kelvin, so at room temperature $k_B\ T\ ln\ 2$ is roughly half the energy of a single Hydrogen bond). In contrast, Feynman's quantum computer can in principle run with essentially zero heat dissipation, since it does not destroy any information.

In the early 1980s, computers could just about perform a few 10^9 floating point operations per second, and each logical operation dissipated around $10^{10}\ k_B\ T$ units of energy—10 orders of magnitude above the Landauer limit. Feynman thus joked: "The question is academic at this time. ... Such nonsense is very entertaining to professors like me. I hope you find it interesting and entertaining also."

His entertaining idea has spawned vast amounts of research over the past few decades, and his particular construction of embedding quantum computation into a physical system remains essentially unchanged to date.

COMPUTATIONAL HISTORY STATE HAMILTONIANS

What Feynman described is now widely known as a so-called "history-state Hamiltonian" H. Referring to the fact that a Hermitian operator H encodes in its lowest eigenvector a superposition over all the time steps of the computation. More formally, we work with a multipartite Hilbert

space $\mathcal{H} = \mathbb{C}^T \otimes (\mathbb{C}^2)^{\otimes n}$, which can be regarded as a clock storing the current position of the computation from 1,...,T, and a state space storing the configuration of n (qu)bits. As common in quantum information, we will use braket notation: Vectors in the Hilbert space will be written as "ket" $|\psi\rangle$, and dual vectors as "bra" $\langle\phi|$. An inner product is then written as $\langle\phi|\psi\rangle$, and the outer—or Kronecker—product $|\psi\rangle\langle\phi|$. Gates are unitary operations acting on this Hilbert space. For a circuit built from the gates $U_1,...,U_T$, we want to construct a Hamiltonian H with ground state $|\Psi\rangle = \Sigma_t |t\rangle \otimes U_t \cdots U_1 |\phi\rangle$, i.e. a uniform superposition over the history of the computation. If $|\phi\rangle$ is some valid initial configuration for the circuit, then the state entangled with the time register encodes a valid computation. More simply put, $\langle t|\Psi\rangle = U_t ... U_1 |\phi\rangle$ encodes the state of the computation at time t. One can show that the Hamiltonian:

$$H = \sum_{t=2}^{T} |t-1\rangle\langle t-1| \otimes id + |t\rangle\langle t| \otimes id - |t\rangle\langle t-1| \otimes U_t - |t-1\rangle\langle t| \otimes U_t^\dagger$$

achieves this. It resembles a lazy random walk along the time direction of the computation: For some state, $|\psi\rangle = |t\rangle \otimes |\phi\rangle$, its action is $H|\psi\rangle = |t\rangle \otimes |\phi\rangle - |t+1\rangle \otimes U_{t+1}|\phi\rangle - |t-1\rangle \otimes U_t^\dagger |\varphi>$, where $U_t^\dagger$ denotes the conjugate transpose of U_t, i.e. the inverse gate. The ground state of H is then the stationary distribution of an unbiased walk on the vertices $\{1,...,t\}$, i.e. $|\Psi\rangle$. The $|\phi\rangle$ can be enforced to be some valid initial configuration for the computation by adding a local Hamiltonian, projecting on invalid states and thus giving them an energy penalty.

This idea led to a series of interesting results. Alexei Kitaev [2] proved deciding whether the ground state of such a Hamiltonian is below some threshold α, or above β, is a complete problem for the complexity class QMA—the quantum analogue of NP (meaning that witness and verifier are a quantum state and quantum circuit, respectively, and accepting and rejecting are probabilistic), even for spin systems with a five-local Hamiltonian. Such systems have overall Hilbert space $H = (C^d)^{\otimes N}$, i.e. a tensor product of dimension d spins, and $H = \sum_i h_i$, where each h_i acts non-trivially on at most five spins at the same time. The thresholds α and β are promised to be separated by $\beta - \alpha = \Omega\left(\frac{1}{poly\,T}\right)$, lower-bounded by some inverse polynomial in the runtime T of the com-

Image by Aleksei Vasilelka

putation. Kitaev's reduction is based on an embedding of a verifier circuit for a QMA-hard decision problem, following the construction outlined above: For a NO instance, the circuit would reject any witness state $|\phi\rangle$ with high probability, and the output bit would have large overlap with a penalty term meant to push the ground state energy above β. For a YES instance, this overlap is small, upper-bounding the ground state energy by α. The problem of distinguishing the ground state energy between those two cases is thus QMA-hard.

These results have been improved successively, and are becoming ever more relevant for real-world systems. In 2009, Gottesman and Irani published a remarkable construction on a spin chain, which remains hard even for translationally invariant nearest-neighbor interactions [3]. Almost all of these constructions embed a standard model of quantum computation, i.e. either a quantum circuit—in which case the operations U_i directly correspond to gates—or a quantum Turing machine. Proposed in 1985 by David Deutsch, quantum Turing machines are universal for quantum computation and can be regarded as an extension of classical Turing machines; where the internal state and tape are replaced by states in a Hilbert space, and the partial transition function becomes a unitary operation on this Hilbert space.

UNDECIDABILITY OF THE SPECTRAL GAP

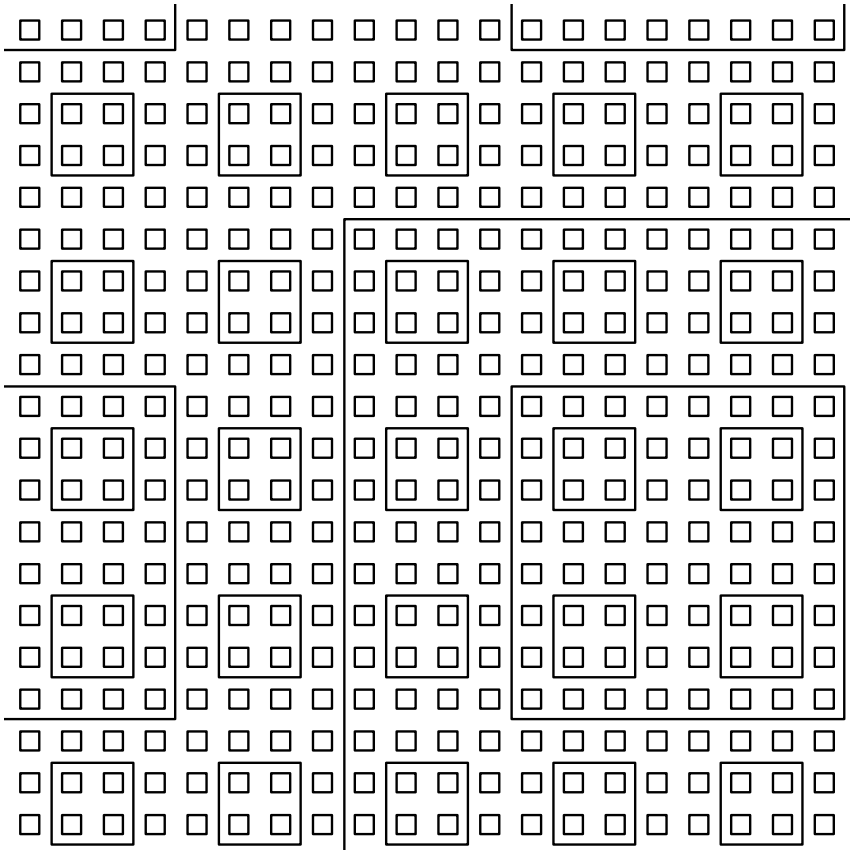
If you followed closely and you are vaguely familiar with computer science and quantum physics, your alarm bells might go off at this point. Couldn't you embed a Turing machine into a Hamiltonian, and prove that physical property of the system it describes is in fact undecidable by reducing it to the halting problem? If instead of penalizing the NO output of a verifier computation we penalize the halting state of a Turing machine, the resulting Hamiltonian will have its low energy ground state above some threshold β if the Turing machine halts, and it will lie below α if the machine runs forever. But this argument suffers from a crucial problem: If the promise gap $\beta - \alpha$ closes inverse-polynomially in the system size, the ground state energy for halting and non-halting will be indistinguishable in the thermodynamic limit, i.e. infinite system size. Whether or not there exists a constant promise gap history-state Hamiltonian is highly nontrivial, and still an open problem. But with a highly technical construction, Cubitt, Perez-Garcia, and Wolf found a way around this obstacle and constructed a physical system for which deciding whether it is gapped or gapless in the thermodynamic limit is undecidable [4]. But extraordinary claims require extraordinary evidence, so let us take a step back at this point and try to understand what exactly the three authors proved.

The spectral gap of a physical system, described by a Hamiltonian H, is defined as the difference between the energies of the first excited state and the ground state, written as $\Delta = \lambda_1(H) - \lambda_0(H)$, where λ_0 and λ_1 denote the two lowest eigenvalues of H. For understanding the behavior of quantum many-body systems in the thermodynamic limit, this spectral gap plays a fundamental role. In condensed matter theory, the gap behavior is tightly linked to the phase diagram of the system, and phase transitions occur at critical points where the gap vanishes. In adiabatic quantum computation, which is as powerful as the circuit model, the minimal spectral gap along the path of Hamiltonians, which are adiabatically tuned, determines whether the computation can be carried out efficiently [5].

Numerous examples show even deciding whether a specific system

Figure 1. Finite section of the quasi-periodic Robinson tiling.

The tile set consists of a finite set of tiles, and the emerging structures are squares, arranged as shown in the picture: Every four squares are contained in a bigger square. For any n, there is therefore a constant nonzero density of structures of size 4^n.



is gapped or gapless is a highly nontrivial task. For the one-dimenstional antiferromagnetic Heisenberg model, this question is known as the Haldane conjecture, and open since 1983 [6]. In high-energy physics, the question whether Yang-Mills theory—a non-abelian gauge field theory describing the strong and weak nuclear force—has a mass gap is one of the Millennium Prize Problems.[1]

Cubitt, Perez-Garcia, and Wolf prove the general problem of deciding whether a physical system is gapped or gapless in the thermodynamic limit is undecidable in the exact same sense as the halting problem for a universal Turing machine is undecidable. Their notion of gapped and gapless is unambiguous: The system they construct either has continuous spectrum above the ground state in the thermodynamic limit, or a unique ground state with a finite, constant gap $\Delta \geq c$ for some constant $c>0$. The system is described by a translationally-invariant, nearest-neighbor Hamiltonian $H^{\Lambda(L)} = \sum_i h_i^{row} + h_i^{col} + h_i$, defined on a 2-D lattice $\Lambda(L)$ of spins with constant (yet large) dimension; where all couplings between neighbors h_i^{row}, h_i^{col} and 1-local terms h_i have coupling strengths $\| h_i \| \leq 1$ with computable (algebraic) matrix entries.

In other words, their findings show there exists no algorithm, however inefficient, which can distinguish the gapped and gapless case in generality—even if restricted to simple systems as just described. In the axiomatic sense, this shows in any consistent recursively-defined formal system that allows this problem to be stated, neither the presence nor the absence of a spectral gap is provable from the axioms.

COMBINING WANG TILES, QUANTUM PHASE ESTIMATION, AND THE HALTING PROBLEM

Cubitt, Perez-Garcia, and Wolf's result is based on a quasi-periodic Wang tiling serving as a base layer, on top of which lives a universal classical Turing machine. The Wang tiling is purely classical, and the Turing machine is written as a history-state Hamiltonian. The input for the universal Turing machine is computed from a phase encoded in a local interaction, which is done by a special-purpose quantum Turing machine performing quantum phase estimation.

The Wang tiling is probably the most intuitive part of the construction, and it is helpful to think of a puzzle as analogy. Imagine you have a finite set of squares W, where each edge is colored from a finite set of colours S. Of each square you have an unlimited supply. You start tiling the plane, requiring adjacent tiles have matching colors. It is straightforward to write down a Hamiltonian H on a square lattice Λ, where each edge carries a spin of dimension $|S|$, and such that the ground state of H corresponds to valid tilings. Denoting neighboring lattice sites with $i \sim j$, we write:

$$H_W = \sum_{i \sim j} \sum_{s_i, s_j \in S} (1 - f(i,j,s_i,s_j)) id \otimes id,$$

where $f(i,j,s_i,s_j) = 1$ if s_i and s_j match on sites i and j, and 0 otherwise. This Hamiltonian acts on nearest neighbor spins only. Each term can be thought of as a stabilizer operator, allowing only valid tile combinations on neighboring edges: Similar to a classical constraint satisfaction problem, a state which is in the kernel of all local terms is thus a solution to the tiling problem.

A quantum Turing machine with bounded state space and alphabet can write out a countably infinite number of outputs, even if only provided with a fixed input.

It has been shown the tiling problem itself—with unbounded number of edge colors—is undecidable [7]. Unfortunately, the local dimension grows with the number of tiles in the set. If one is not concerned with a bounded local dimension, this is already enough to prove undecidability of the spectral gap. From a physical perspective, however, it is unreasonable to allow the spin dimension to grow without bounds, so one has to find another way to specify the countably infinite number of problem instances necessary to prove undecidability.

Cubitt et al. solve this by using their only inherently quantum ingredient. While a classical Turing machine with bounded state and alphabet size cannot write out a countably infinite number of outputs if provided with a fixed input, a quantum Turing machine can. Its entries, which are complex-valued, can have arbitrary precision as they are specified by a unitary transition matrix. The authors thus explicitly construct a special-purpose quantum Turing machine that performs phase estimation, writing out the entire binary expansion of such a coefficient. It is known that classical probabilistic Turing machines can be used to write out an estimate of the binary expansion of a coin bias. What might come as a surprise is a quantum Turing machine can achieve this deterministically. This distinction is crucial: If the algorithm worked only up to some high probability, the proof would not go through. (The phase estimation Turing machine is exact given that it can run for long enough to write out the entire phase. On tapes that are too short, the output can be some arbitrary quantum state. The construction is robust to this.)

Combining Wang tiling, phase estimation, and a Gottesman-Irani-style universal Turing machine, Cubitt et al. succeed in proving the ground state energy density of their constructed family of Hamiltonians is undecidable. More specifically, they use a specific tile set [8], which can tile the plane in an aperiodic fashion and exhibits a constant nonzero density of boundaries of all length scales, as illustrated in Figure 1. With the Robinson tiling as a

1 http://bit.ly/2bcHTTF

base layer, they put a phase estimation quantum Turing machine dovetailed by the universal Turing machine on every lower square edge. If the universal Turing machine halts after T steps on the input provided by the phase estimation, it will pick up a penalty p of size $O\left(\frac{1}{poly\,T}\right)$ for every copy of the Turing machine running on a boundary large enough to allow the Turing machine to run for T steps. As the halting time for a universal Turing machine is uncomputable, this penalty will be uncomputably small. However, it is nonzero and constant, and in particular independent of the overall size of the system. On the other hand, as there are $O(L^2)$ copies of this Turing machine running in parallel on a lattice of side length L, the overall energy penalty will scale as $O(pL^2)$, diverging in the thermodynamic limit. The true analysis is therefore subtle, as this state cannot be the ground state in the limit. It might be favorable to break the Robinson tiling instead, introducing defects. But the authors succeed in showing that their construction is robust to these perturbations, and the energy still scales as L^2 in the system size.

Their resulting Hamiltonian $H_u^{\Lambda(L)}$ has the property that either its ground state energy density is strictly positive—if the Turing machine halts—or approaches zero from below. By combining $H_u^{\Lambda(L)}$ with a gapless local Hamiltonian H_d with zero energy ground state, such that the resulting interaction has one extra non-degenerate zero energy ground state and such that the spectrum of H_d is shifted up by $\lambda_{min}(H_u^{\Lambda(L)})$, the resulting Hamiltonian will be gapped—revealing the extra zero energy stvate—if and only if the Turing machine halts on the given input. This gap is constant, and lower-bounded by one, as $\lambda_{min}(H_u^{\Lambda(L)}) \to \infty$ as $L \to \infty$, and their claim follows.

CONCLUDING REMARKS

The implications of this result for condensed matter physics are profound. It demonstrates one can write down a family of relatively simple physical systems with phase diagrams that are uncomputably complicated. Any attempt at solving the system for ever larger instances can only give misleading results: Adding a single row of atoms could completely change the system from gapped to gapless. From a mathematical perspective, a more striking statement is the logical impossibility of giving a full characterization of the gapped and gapless Hamiltonians. This statement is much stronger than merely stating that it is numerically intractable. For specific instances, e.g. the spin-1 antiferromagnetic Heisenberg model, the question of whether it is gapped or not is trivially decidable. However, for the general case, we will never be able to write down a classification of Hamiltonians distinguishing gapped and non-gapped cases.

The idea of uncomputable or undecidable quantities is not new to physics, and there are other results—albeit much easier ones, e.g. by relaxing the condition on translational invariance or geometric locality. In light of the significant number of highly non-trivial obstacles that had to be overcome in the analysis of their construction—only a fraction of which could be outlined in this summary—Cubitt et al.'s undecidability result can be seen as a major contribution to our understanding of spectral properties of many-body quantum systems.

Since the paper was published in *Nature* in 2015, there have been a few interesting results related to this line of work. Closely linked is a contribution from Bausch et. al., which proposes a simple, physically realistic model on a lattice exhibiting a "phase transition" at growing system sizes [9]. Switching from purely classical ground and first excited states to exhibiting topological order with anionic excitations as in Kitaev's Toric Code, the system becomes quantum when it is larger than some threshold lattice size. This is counter intuitive, and could open up possible applications for materials with exotic properties.

Moreover, one could imagine combining the Hamiltonian construction with other interesting computational problems. In a more recent paper, the authors construct a Turing machine for which the halting problem is independent of ZFC set theory [10]. Embedded in a similar fashion as in the undecidability case, this would show that deciding whether the resulting Hamiltonian is gapless is not possible within ZFC, assuming it is consistent. Other variants would allow constructing Hamiltonians that are gapped if and only if the Riemann Hypothesis—or Goldbach's Conjecture—is correct. As Feynman said, these lines of thought are academic at this point. It remains to be seen if we will gain deeper insight from them, but I hope some will find them entertaining and interesting as well.

References

[1] Feynman, R. P. Quantum mechanical computers. *Foundations of Physics* 16, 6 (1986), 507–531.

[2] Kitaev, A. Y., Shen, A. and Vyalyi, M. N. In *Quantum Information*. Springer, New York, 2002, 203–217. doi: 10.1007/978-0-387-36944-0_13.

[3] Gottesman, D. and Irani, S. The quantum and classical complexity of translationally invariant tiling and Hamiltonian problems. *Theory Comput.* 9 (2013), 31–116.

[4] Cubitt, *T.* S., Perez-Garcia, D. and Wolf, M. M. Undecidability of the spectral gap. *Nature* 528, 7581 (2015), 207–211.

[5] Farhi, E., Goldstone, J., Gutmann, S. and Sipser, M. Quantum Computation by adiabatic evolution. arXiv. 2000; http://arxiv.org/abs/quant-ph/0001106/.

[6] Haldane, F. D. M. Nonlinear Field theory of large-spin Heisenberg antiferromagnets: semiclassically quantized solitons of the one-dimensional easy-axis Néel state. *Physical Review Letters* 50 (1983), 1153–1156.

[7] Berger, R. *The Undecidability of the Domino Problem*. American Mathematical Society, Providence, 1966.

[8] Robinson, R. M. Undecidability and nonperiodicity for tilings of the plane. *Inventiones mathematicae* 12, 3 (1971), 177–209.

[9] Bausch, J., Cubitt, *T.* S., Lucia, A., Perez-Garcia, D. and Wolf, M. M. Size-driven quantum phase transitions. arXiv. 2015; http://arxiv.org/abs/1512.05687/.

[10] Yedidia, A. and Aaronson, S. A relatively small Turing machine whose behavior is independent of set theory. arXiv. 2016; http://arxiv.org/abs/1605.04343..

Biography

Johannes Bausch is a third year Ph.D. student at the Centre for Quantum Information and Foundations at DAMTP in Cambridge, UK. He is particularly interested in Hamiltonian complexity theory and quantum stochastic processes.

PROFILE DEPARTMENT EDITOR, ADRIAN SCOICĂ

David Deutsch
Understanding Computation as a Consequence of Physics

DOI: 10.1145/2983453

David Deutsch is currently a visiting professor of physics at Oxford University. He is widely regarded to have laid the theoretical foundations for the theory of quantum computation by being the first person to describe a universal quantum computer. However, although his insight into physics and quantum computing is well known and self-evident from his fruitful scientific output, the background life stories against which his observations arose and evolved are perhaps less familiar to the community.

Herein Prof. Deutsch offers a more personal account of his work. It is through this account that I pleasantly discovered him to be a humble and engaging conversationalist. I hope you, the reader, will be inspired to go away and personally explore some of the same mental paths through which he successfully ended up challenging the status quo of physics in his day.

"I'LL GET INTO PHYSICS LATER..."

David Deutsch was raised in the UK after his family immigrated from Haifa, Israel when he was only three years old. According to his own account, Deutsch cannot remember how or when specifically he knew he was interested in science, but on an instinctive level he possessed the curiosity. It was on the first day at a new school at the age of 11, he finally learned what physics was. In fact, physics was not even his first passion in school. Back in those early days, his favorite subject was the more granular and empirically-formulated subject of chemistry. He recalls having had a conversation with a friend around the age of 14 during which he said: "I'm kind of obsessed with chemistry at the moment, but pretty soon at A-levels[1] I'll get more interested in physics." True to his word, he consequently did.

Deutsch went on to complete a bachelor's degree in natural sciences at the University of Cambridge. It was during this time he became unhappy with the explanations he read. It dawned on him that in order to think outside the box and understand more about the laws of nature one has to look beyond science as a prefabricated construct, and instead look into the philosophy of science and the relationship of theory to experiment. As he was struggling to formulate his point of view, Deutsch accidentally discovered, with the help of a historian tutor, the work of the then-poorly known philosopher Karl Popper. Deutsch attributes his later critical and very successful approach to rationalizing physics to the transformative power of Popper's ideas.

At the end of his degree, Deutsch stayed on at Cambridge for a Part III in mathematics (a master's course), which he amusingly referred to as likely a mistake. "I didn't really plan my career, ever. I just did what I found was interesting, and from time to time I sort of applied to things in order to find ways of doing what I wanted to do," he told me, adding he found the Part III to be boring. As a result, he did not attend many of the courses and did poorly.

With his time in Cambridge drawing to an end, Deutsch began looking for interesting things to do next, which led him to apply for a DPhil position at Oxford with Prof. Dennis Sciama. His initial thesis proposal was on quantum gravity; perhaps another self-confessed mistake, since he did not know enough at the time about either quantum theory or relativity in order to internalize the problem of unifying them. "In those days, you didn't have to stick to your thesis topic; you could always change it later. I should have actually described my topic in terms of problems that I had, not problems that I thought physics had. But I did want to understand these two topics individually, and that's what I did when I started research. Luckily, Dennis Sciama was a very accommodating boss, and his view was that as long as I was engaged with my work, something would come out of it. And things slowly did...," he recalled.

"WELL, THEN, YOU'RE USING THE WRONG PHYSICS!"

As most aspects of life, the story of how Deutsch became interested in quantum computing also lies under the auspices of serendipity, and came about in a convoluted way, emerging from his early interest in the nature of explanations and the seemingly lacking character of the explanations of his day. "I was interested in the philosophy of science, and what counts as a good theory in science. In regard to quantum theory, this is a very subtle issue. And I was dissatisfied with the way physicists accepted the silliness of wave function collapse, which to me seemed like an abdication of the entire purpose of science—which is to understand the world. It was almost a 'you can't ask that question' sort of attitude, or 'shut up and calculate', as it's now called," he explained. Furthermore, the approach to quantum mechanics he read struck him as being too operational.

In his quest for a more fundamental explanation than the wave function

1 The A-levels in the UK are qualifications offered at the end of high school, similar to the Baccalauréat or Abitur in mainland Europe, or the SAT in the USA.

collapse theory during his early days in Oxford, Deutsch came across the Everett interpretation of quantum mechanics. After reading all related papers (only about half a dozen at that time), he came to realize they were all wrong on one detail: There was no way to distinguish between the Everett many-world theory and the wave function collapse theory. He proposed a thought experiment to distinguish between the two, which required the existence of an entity that would count as an observer and yet would maintain quantum coherence.

That entity is what we would today call a quantum computer. The interpretation, however, eluded him at the time, and he filed the paper away in favor of other work. His later Ph.D. years took him to Texas, where he had a conversation with Charles Bennet of IBM Research during a conference. Unaware that Bennett had been among the people who developed complexity theory in the first place, Deutsch professed his belief that complexity theory was rubbish because it was ultimately dependent on the hardware: Different hardware would give you different complexities. Bennett pushed back by pointing out complexity theory is a rigorous science because, ultimately, the hardware is physics itself. At that point, it dawned on Deutsch that they should have built up their theory using quantum physics. He then went back to redo Turing's argument in terms of quantum physics instead of classical physics. He initially did not expect much to come out of it, and yet that was the beginning of the quantum theory of computation.

COMPUTATION AS A MANIFESTATION OF PHYSICS

The connection between computation and physics, Deutsch explains, is more fundamental than the horizon-limiting applications of the theories might suggest. Babbage's first thought of classical computation, for example, was in terms of its applications. He designed the first brass machine in order to make mathematical tables not subject to human error, but he did not think beyond applications to

Photo by Lulie Tanett © David Deutsch 2008.

mathematical calculation. It was Ada Lovelace who, upon contemplating the machine, realized this proposed computer of Babbage's could simulate any physical process, and the set of motions of this analytical engine was isomorphic to the set of possible motions of any object: The whole universe encodable in a single machine.

The fact that such a machine can even exist in the first place is a very deep property of the laws of [classical] physics.

Now, with quantum physics, it turns out a Turing machine was not the right object with that property, and the object with that universality property is instead a universal quantum computer. What is significant here is the connection between computational information and physics continues to hold, and is fundamental.

"I REALLY HATED APPLYING FOR RESEARCH GRANTS"

During our interview, Deutsch confessed even though he loved being in a university environment, he had a core dislike of the administrative and bureaucratic side of living in the university system. He especially disliked being a lecturer because he hated the idea of giving a talk to people who aren't there because they want to understand, but rather because they had another purpose (such as meeting a requirement or passing an exam).

As such, he fought to forge an alternative path for himself, and through a complicated process, he ended up being commissioned to write two books—*The Fabric of Reality* and *The Beginning of Infinity*—which were quite successful.

Not only did writing books help free him from academic bureaucracy, but it also enabled him to write down his ideas in a more structured way that would efficiently record them for posterity. However, he made it clear to me that in his intellectual endeavors, persuading others he is right was never a top priority. In that same spirit, he regards having been elected a Fellow to the Royal Society as personally unremarkable: "I'm glad they did it, because it shows my take on things is, in some respect, beginning to catch on, but apart from that it's not a very big deal. I'm not really interested in career."

Instead, Deutsch is happy to point out the biggest reward in life for him comes from enjoying physics itself: "I think as a physicist, every day is happy... and equally happy!"

end

LABZ

UC Berkeley's Quantum Computing Group

Berkeley, CA

Editor's Note: *Members of the UC Berkeley quantum computing group are devoted to understanding the enormous power of quantum systems. What will the computers of tomorrow bring? Seung Woo Shin attempts to answer that question.*

Quantum computing could revolutionize the world of computing, and, not surprisingly, has received a lot of attention from both academia and the public over the years. As Richard Feynman observed, quantum mechanics is complex. The number of parameters required to describe a quantum state grows exponentially in the number of particles, as opposed to linearly as in the classical case. But this exponential complexity of quantum mechanics is a fundamental computational resource, which allows the quantum computer to solve certain problems (e.g. factoring) exponentially faster than the classical computer. On the other hand, this exponential complexity also presents a severe challenge to researchers working in related fields. Namely it means researchers need to understand and engineer systems that could be exponentially more powerful and expressive than themselves.

At UC Berkeley, the quantum computing group studies various issues surrounding the exponential nature of quantum systems. Led by Professor Umesh Vazirani—one of the founders of quantum computing—the group works not only on more traditional topics, such as quantum algorithms and quantum complexity theory, but also more recent topics such as quantum Hamiltonian complexity. Naturally, we collaborate with researchers from a wide spectrum of disciplines. For example, in 2014 the semester-long program at UC Berkeley's Simons Institute brought together more than 50 researchers worldwide from computer science, mathematics, and physics to explore the rich connection between quantum information science and condensed matter physics.

The last few years have produced many exciting engineering advances in quantum technologies. The most famous of these would be the large-scale implementation of quantum annealers by D-Wave Systems. While falling short of achieving general-purpose quantum computation, these machines provide a novel heuristic for solving certain optimization problems (e.g. minimizing the energy of a classical spin system), and may have the potential to achieve a quantum speedup on those problems. Unfortunately, in the absence of a rigorous protocol for testing such machines, their exact computational power remains unknown. The difficulty stems from the fact that quantum computers are supposed to be exponentially complex and therefore cannot be simulated on a classical computer. How, then, can we test that a given quantum computer actually behaves according to specification, or even that it is indeed "quantum"?

2^{1000} **The number of search possibilities using D-Wave Systems' quantum computer of 1,000 qubits. This figure is higher than the total number of particles in the universe.**

At Berkeley, my research primarily consists of developing an appropriate benchmark for the quantum annealer. The main idea of our method is borrowed from the famous Turing test, which compares the black-box behavior of a given machine with that of humans in order to determine whether the machine can "think." Similarly we compare the black-box behavior of a given quantum computer to that of a suitable classical model, in order to test whether the quantum computer exhibits nontrivial quantum effects and whether it achieves a quantum speedup. If the two are nearly indistinguishable, the quantum computer is arguably "classical" from a computational viewpoint.

So far, our tests have not exhibited strong evidence that today's quantum annealers are capable of inducing nontrivial quantum effects. This means experimental data from these machines, which are publicly available, can mostly be explained using our classical model. While this suggests we still have a long way to go in quantum engineering, our methodology for testing quantum annealers is already impacting follow-up research in various ways. Our work not only provides an effective benchmark with which to understand where our technologies stand, but it also provides a guideline as to where to look for quantum effects in these machines. All in all, it is my great fortune to be able to contribute to the development of this exciting technology as a member of UC Berkeley quantum computing group, and I hope our work will continue to play an important role in bringing the quantum era into reality.

Biography

Seung Woo Shin is a graduating Ph.D. student in computer science at UC Berkeley, advised by Umesh Vazirani. His research focuses on the question of how we as classical beings can understand and control quantum systems which may be exponentially more powerful than ourselves.

BACK

The RSA Trap

The Rivest-Shamir-Adleman (RSA) cryptosystem is one of the most commonly used public-key encryption algorithms. The public keys are publicly disseminated, so that anyone sending a message to a party A can encrypt the message using A's public key. But the message can only be decrypted with A's private key, which rest solely in the possession of the rightful addressee of the message. The algorithm works by finding a large semiprime n such that $n = p \times q$. The values (n, p, q) are used to generate a public key that's composed of n, a public exponent and a private key that is composed of n, and a private exponent. To be able to decipher a message given n, one needs to factorize n value to find p and q values.

In 1991, RSA Laboratories set out to learn the effort required to factor RSA numbers of a given size, hence the RSA Factoring Challenge. The challenge became inactive in 2007. To date the largest semiprime factored was the 768-bit challenge in 2009. As the semiprime grows bigger, the factorization time grows exponentially, which evokes the need for a quantum computer.

In 1994, Peter Shor formulated a quantum integer factorization algorithm to factorize any number in theory, which has been recently realized practically in 2016 using a scalable ion-trap quantum computer that returns the correct factors of the number 15 with a confidence level exceeding 90 percent. Researchers claim this computer can be scaled to factor larger numbers in a polynomial time, which would jeopardize RSA's security.

—Asmaa Rabie

Quantum Factorization Records of Semiprimes

Number	Qubits	Algorithm	Year Broken	Scalable
15	8	Shor	2001	No
21	10	Shor	2012	No
56153	4	minimization	2012	No
15	7+4 (cache-qubits)	Shor	2016	YES

Image by Fatmawati Achmad Zaenuri/Shutterstock.com

HELLO WORLD

The Infinite Mixtures of Food Products

BY MARINKA ZITNIK

Imagine you are a data-curious foodie who plans to organize a birthday party for a good friend of yours. Your goal is to find natural groups of food products so you can better cater to the tastes of your friends. For example, your vegan friends might love soy burgers and brown rice marshmallow treats, your carnivore friends might love steak tartare and sashimi, and your healthy eating friends might enjoy a harvest salad and Mediterranean panini.

One way to tackle this task is to make a large list of food products and then analyze their nutrition facts and lists of ingredients. A simple approach involves using a finite mixture model [1] to cluster the products such that foods in the same cluster have similar nutritional values and contain similar ingredients. This is sometimes called model-based clustering because it involves defining a probabilistic model of the data and optimizing a well-defined objective, such as the likelihood or posterior probability of the data [1].

The issue with finite mixture models is that the models assume a finite and fixed number of clusters in the data, which has to be specified in advance before the analysis is started. However, in many cases, there is no well-defined number of clusters and it is not clear if the "correct" value for the number of clusters even exists. In our case, there really might be an infinite number of food tastes and any list of food products, though long, might simply not have enough data to detect all the different tastes. We would thus like to have a model that posits an infinite number of food clusters, which naturally emerge when analyzing longer lists of food products. Your foodie birthday party planning would be much more convenient if you did not have to choose the number of clusters at all.

In this column, we discuss infinite mixture models, which do not impose any a priori bound on the number of clusters in the data. To be able to achieve this level of flexibility, we will use a nonparametric Gaussian mixture model based on the Dirichlet process [1,2]. This will allow the number of clusters to automatically increase as new food products are added to the list.

THE CHINESE RESTAURANT PROCESS

Let us describe the infinite mixture model for finding food clusters as the following generative story. We assume an infinite set of clusters, where each cluster is described by a set of parameters. For example, in the analysis presented here, each cluster is described by a Gaussian distribution with a specified mean and a standard deviation. These cluster parameters themselves come from another distribution, which is often called a base distribution. One of the canonical generative stories for the infinite mixture models is called the Chinese restaurant process [1 ,2, 3], which we use here to assign food products to clusters.

The Chinese restaurant process works as follows. Imagine a Chinese restaurant where all your friends go to eat one day. Initially, the restaurant is empty. When the first person enters the restaurant, she sits down at a table and orders food for the table; everyone else who joins her table will be limited to eating the food she ordered. In the modeling speak, sitting down at a table corresponds to selecting a cluster and ordering the food corresponds to selecting parameters for the cluster. Next, the second friend enters the restaurant and he has to pick a table to sit at. With probability $\alpha/(1+\alpha)$ he sits down at a new table (i.e., selects a new cluster) and orders food for the table (i.e., specifies parameters for the new cluster). Otherwise, with probability $1/(1+\alpha)$ he joins the table of the first person and eats the food that is already ordered (i.e., he and the first person belong to the same cluster). After some time, the $(n+1)$-st friend enters the restaurant and sits down at a new table with probability $\alpha/(n+\alpha)$ and at table k with probability $n_k/(n+\alpha)$, where n_k is the number of your friends currently sitting at table k.

Although the Chinese restaurant process might at first sight appear a very simple model, it is an extremely powerful concept [1, 2, 3] that is currently an active research topic and has seen many applications [4, 5]. For example, it is interesting to see the more people (i.e., data points) there are at a table (i.e., cluster), the more likely it is people (i.e., new data points) will join the table. This means that our clusters satisfy a "rich get richer" property. Second, there is always a small probability that someone joins an entirely new table (i.e., a new cluster is formed). And thirdly, we see the probability of a new cluster depends on the value of parameter α. We can think of α as a concentration parameter that affects the dispersion of people in the restaurant. Smaller values of α result in more tightly clustered data points, whereas larger parameter values indicate that for any given finite set of points more clusters will be non-empty.

THE OPEN FOOD FACTS

The Open Food Facts (http://world.openfoodfacts.org) is a collaborative, free, and open database of food products from around the world. For the purpose of this column we retrieved data for 2,776 food products sold in the United States. Examples of the products include various brands of peanut butter, chocolate, bread, meat, and foods from other categories. For each product, we retrieved a list of ingredients together with the associated quantities, nutrition facts, and characteristics of the product.

We represent each product with a vector of real values, such that categorical items are encoded with binary features and numerical items are normalized to report values for same product size (i.e., 100 grams or 100 milliliters). Additionally, we remove features with low variance and food products with empty data profiles. (The raw and preprocessed data sets are available in the supplementary materials.)

CLUSTERING THE FOOD FACTS

Having constructed food profiles based on data from the Open Food Facts database

Figure 1. A cluster of food products discovered by mining the Open Food Facts data. Nutritional profile of the cluster shows that foods in the cluster are low in carbohydrates, fiber, sugars, proteins, calcium and iron, but high in energy and fat (left). Examples of products assigned to this cluster are well aligned with the nutritional profile of the cluster (right).

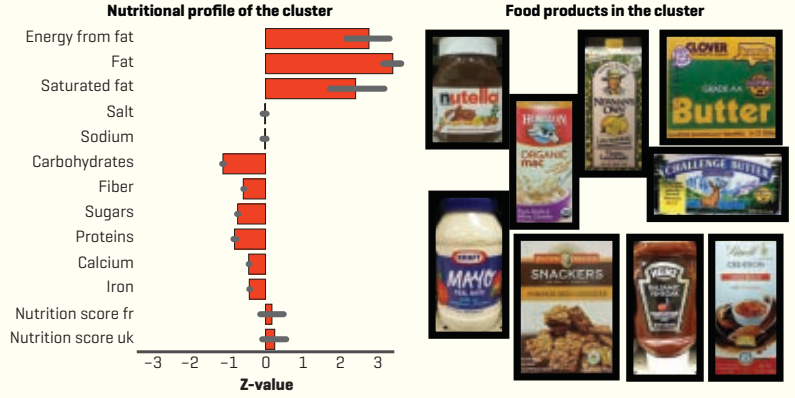
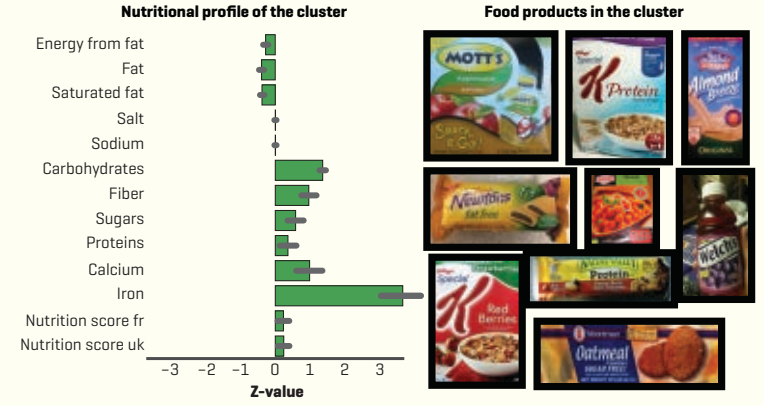


Figure 2: A cluster of food products found in the Open Food Facts data by the infinite Gaussian mixture model. Nutritionally, the cluster contains food products that are high in carbohydrates, fiber, sugars, proteins, calcium and iron.



we proceed by clustering the profiles into coherent groups of foods. For that we consider an infinite Gaussian mixture model [6]. This is an implementation of the Chinese restaurant process described earlier that allows us to calculate the probability of any particular set of cluster assignments to food products. To learn a good set of such cluster assignments we rely on a popular Markov chain Monte Carlo (MCMC) approach, known as Gibbs sampling [1], which is the MCMC analog of coordinate descent. (The Python implementation of the method is provided in the supplementary materials.)

Most importantly, since we are using an infinite mixture model, we do not need to specify the number of clusters to find. We can see the number of clusters detected by the model varies as we feed in more food products. As expected, the model discovers more and more clusters as more and more food products arrive. It can be shown that the number of discovered clusters grows logarithmically as more data points are considered [1].

Considering the entire data set, Figures 1 and 2 show two detected clusters. Looking at a sample of food products from the first cluster (Figure 1), we find a lot of desserts and foods that feature butter. We also show the average nutritional profile of the products assigned to the cluster. We define a nutritional profile of a product by normalizing each nutrition feature value with respect to the mean and standard deviation of the feature. The nutritional profile of a cluster is then reported as a profile of standardized scores, also known as z-values. These scores measure how many standard deviations away from the average feature value is a particular value. For example, food products in the cluster from Figure 1 tend to be high in fat and energy, and low in fiber, proteins, and carbohydrates. On the other hand, cluster in Figure 2 contains cereals, oatmeal, and a variety of protein bars, which are aligned with the nutritional profile of the cluster. The profile tells us that this cluster is high in iron, calcium, fiber and proteins, and is low in fat and salt.

Altogether, our analysis here demonstrates that modern statistical methods can find high quality partitioning of the data into clusters without the need to a priori define the number of desired clusters. Furthermore, we surely satisfied data curiosity of our foodie friend!

References

[1] Murphy, K. P. *Machine learning: a probabilistic perspective*. MIT Press, Cambridge, 2012.

[2] Rasmussen, C. E. The infinite Gaussian mixture model. In *NIPS*, 1999, 554-560.

[3] Yerebakan, H. Z., Rajwa, B., and Dundar, M. The infinite mixture of infinite Gaussian mixtures. In *NIPS*, 2014, 28-36.

[4] Liu, C. L., Tsai, T. H., and Lee, C. H. Online Chinese restaurant process. In *ACM SIGKDD*, 2014, 591-600.

[5] Kyung, M., Gill, J., and Casella, G. New findings from terrorism data: Dirichlet process random-effects models for latent groups. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 60, 5 (2011). 701-721.

[6] Kamper, H., Jansen, A., King, S., and Goldwater, S. Unsupervised lexical clustering of speech segments using fixed-dimensional acoustic embeddings. In *Spoken Language Technology Workshop (SLT)*, 2014, 100-105.

Supplementary Materials

http://github.com/acmxrds/fall-2016

Biography

Marinka Zitnik is a Ph.D. student in computer science at the University of Ljubljana. She has also completed research at the University of Toronto, Imperial College London, Baylor College of Medicine, and Stanford University. Her interests include machine learning, artificial intelligence, probabilistic numeric, and bioinformatics.

Images by The Open Food Facts, http://openfoodfacts.org

ACRONYMS

EPR Paradox Einstein-Podolsky-Rosen Paradox: The EPR Paradox was a quantum mechanics experiment done by Einstein and his two associates, Boris Podolsky and Nathan Rosen. The fundamental principle of this paradox is the interaction of particles happens in such a manner, that it's possible both their momentum and position can be measured more precisely as compared to the Heisenberg's uncertainty principle, provided that the measurement of one particle does not affect the other one instantaneously to prevent it.

Qubit Quantum Bit: A qubit, or quantum bit, is a quantum computing counterpart to the bit or a binary digit of classical computing. In classical computers, a bit represents the elemental unit of information. Similarly in a quantum computer, a qubit is the basic unit.

QBER Quantum Bit Error Rate: It is stated QBER is the ratio between the number of bits in error and total bits detected. Mostly, analysis of QBER is focused on a particular component's effect in the link. Dark count refers to total error occurrence due to dispersion, detectors, imperfect sources, and loss.

QKD Quantum Key Distribution: Security in communication is guaranteed by QKD through quantum mechanics. Two parties create a random shared secret-key that is only known by those parties. Messages can be encrypted and decrypted using this secret-key.

POINTERS

QUANTUM COMPUTING

Quantum mechanics is a conceptually counterintuitive area of science that has baffled some of the finest minds. Albert Einstein said, "God does not play dice with the universe" when speaking of quantum phenomenon. A quantum computer taps directly into the fundamental fabric of reality—the strange and counterintuitive world of quantum mechanics—to speed up computation. Rather than store information as zeroes or ones as conventional computers do, a quantum computer uses qubits, which can be a one, a zerp, or both at the same time. This "quantum superposition," along with the quantum effects of entanglement and quantum tunnelling, enable quantum computers to consider and manipulate all combinations of bits simultaneously—making quantum computation powerful and fast. In the near future, it will likely become possible to perform special-purpose quantum computations that, while not immediately useful for anything, are plausibly hard to simulate using a classical computer.

—Tejas S. Khot

QUANTUM SUPREMACY

"Has the Age of Quantum Computing Arrived?"
By Andrew Anthony
Computer science has witnessed some astounding developments over the last few decades, but the next anticipated step may be the most revolutionary of all. *The Guardian*'s Andrew Anthony surveys some of the physical implementations of quantum computers, and how they are perceived by the industry from a commercial standpoint.
https://www.theguardian.com/technology/2016/may/22/age-of-quantum-computing-d-wave

"Primitive Quantum Computers ay Already Outperform Standard Machines for Very Specific Tasks"
Present-day quantum computers have a very limited number qubits at their disposal owing to the fragile nature of quantum entanglement, which makes it extremely difficult to maintain many more of them. Even with these constrained quantum processors, researchers are able to achieve higher performance for solving extremely specialized problems. Researchers are studying the processing capabilities of smaller, simpler designs as precursors for large-scale quantum computers and also as processing units in their own right. The upshot of this research is a new link has been established between quantum walks and computational complexity theory that shows specific tasks could ultimately demonstrate quantum supremacy over classical computers.
http://www.gizmag.com/quantum-computer-processor-walk-algorithm/43263/

"Strachey Lecture : Quantum Supremacy"
By Dr. Scott Aaronson
On what grounds should we believe that a given quantum system really is hard to simulate classically? Does classical simulation become easier as a quantum system becomes noisier? And how do we verify the results of such an experiment? In this timely video lecture held at Oxford, Dr. Aaronson of MIT and UT Austin discusses recent results and open problems around these questions, using three proposed "quantum supremacy experiments" as examples: BosonSampling, IQP/commuting Hamiltonians, and random quantum circuits.
https://podcasts.ox.ac.uk/strachey-lecture-quantum-supremacy

"An Introduction to Quantum Machine Learning"
By M. Schuld, I. Sinayskiy, F. Petruccione
The advent of modern machine learning has ushered in rapid advances in the classification and interpretation of large data sets, sparking a revolution in areas such as computer vision and natural language processing. Much of our current understanding of the techniques that underlie this revolution owes a great debt to insights first gleaned from condensed matter and statistical physics. Further insights remain to be found at the intersection of machine learning and fields such as statistical physics, condensed matter, and quantum information. This article gives a systematic overview of the emerging field of quantum machine learning, presenting the approaches as well as technical details in an accessible way, while also discussing the potential of a future theory of quantum learning.
https://arxiv.org/abs/1409.3097

"Philosophical Aspects of Quantum Information Theory"
By Christopher G Timpson
For philosophers, and for those interested in the foundations of quantum mechanics, quantum information theory makes a natural and illuminating object of study for a simple reason. One can cast its central concerns in terms of a long-familiar question: "How does the quantum world differ from the classical one?" This paper reviews some of these philosophical aspects of quantum information theory.
https://arxiv.org/abs/quant-ph/0611187

Quantum Computing and Quantum Chemistry
Even the most powerful classical computers struggle when trying to calculate how molecules will interact in a chemical reaction. That's partly because the complexity of such systems doubles with the addition of every atom, as each atom is entangled with all the others. Quantum computers may prove their worth soon in fields like quantum chemistry. Simulating a water molecule, for example, would require only 14 qubits (by contrast, a classical computer simulation of water needs 2^{14} bits). Two contributions describe some of the recent progress at the intersection of these fields:
• Sabre Kais' "Introduction to Quantum Information and Computation For Chemistry"; *http://www.chem.purdue.edu/kais/paper/QICC_Vol%20154_Chap1.pdf*
• James Whitfield's dissertation "At the Intersection of Quantum Computing and Quantum Chemistry"; *http://www.jdwhitfield.com/jdw.thesis.pdf*

INTRODUCTORY BOOKS FOR BEGINNERS

The multidisciplinary field of quantum computing strives to exploit some of the uncanny aspects of quantum mechanics to expand our computational horizons. Collected here are some of the best reads, not demanding advanced prior knowledge, for learning more.

Quantum Computing for Computer Scientists
Noson S. Yanofsky and Mirco A. Mannucci, Cambridge University Press (2008)
Publisher's Description: "With chapters on computer architecture, algorithms, programming languages, theoretical computer science, cryptography, information theory, and hardware, this text has step-by-step examples, more than 200 exercises with solutions, and programming drills that bring the ideas of quantum computing alive for today's computer science students and researchers."

Quantum Computing: A Gentle Introduction
Eleanor G. Rieffel and Wolfgang H. Polak, MIT Press (2014)
Publisher's Description: "With its careful development of concepts and thorough explanations, Rieffel and Polak make quantum computing accessible to students and professionals in mathematics, computer science, and engineering. A reader with no prior knowledge of quantum physics (but with sufficient knowledge of linear algebra) will be able to gain a fluent understanding by working through the book."

Quantum Computing Since Democritus
Scott Aaronson, Cambridge University Press; 1st edition (April 29, 2013)
Publisher's Description: "Written by noted quantum computing theorist Scott Aaronson, this book takes readers on a tour through some of the deepest ideas of maths, computer science, and physics. Full of insights, arguments, and philosophical perspectives, the book covers an amazing array of topics. There are also extended discussions about time travel, Newcomb's Paradox, the anthropic principle and the views of Roger Penrose. Aaronson's informal style makes this fascinating book accessible to readers with scientific backgrounds, as well as students and researchers working in physics, computer science, mathematics, and philosophy."

FEATURED EVENT

Theory of Quantum Computation, Communication and Cryptography (TQC 2016)

Berlin, Germany
September 27-29, 2016

Quantum computing encompasses a wide range of topics like quantum complexity theory, communication, cryptography, coding theory, etc. Research in quantum computation has lead to rapid theoretical and practical improvements in these areas. As we increasingly rely on services in the cyber world, security has become a key issue. Advancements in quantum computation can only add to the problems that researchers need to figure out how to solve.

TQC 2016 focuses on the theoretical aspects of these problems. The conference will bring together researchers so they can interact and share possible problems and solutions with each other. TQC 2016 will also include invited and contributed talks, and a poster session.

Visitors to the conference can also explore the city of Berlin, which has numerous points of interest like Checkpoint Charlie, the Berlin Wall, and Brandenburg Gate.

For more information about the conference, please visit *http://tqc2016.physik.fu-berlin.de/.*

—Darshit Patel

EVENTS

CONFERENCES

QCrypt 2016
Washington D.C.
September 12-16, 2016
http://2016.qcrypt.net/

IEEE HEPC 2016
Westin Hotel
Waltham, MA
September 13-15, 2016
http://www.ieee-hpec.org/

PPSN 2016
Edinburgh, Scotland
September 17-21, 2016
http://www.ppsn2016.org/conference/

ETSI/IQC Workshop on Quantum Safe Cryptography
Toronto, Canada
September 19-21, 2016
http://www.etsi.org/news-events/events/1072-ws-on-quantumsafe-2016/

IQIS 2016
University of Rome La Sapienza
Rome, Italy
September 20-23, 2016
http://www.picque.eu/iqis2016/

TQC 2016
Magnus House
Berlin, Germany
September 27-29, 2016
http://tqc2016.physik.fu-berlin.de/home/

ACM NanoCom 2016
New York, NY
September 28-30, 2016
http://nanocom.acm.org/

IEEE FOCS 2016
New Brunswick, NJ
October 9-11, 2016
http://www.wisdom.weizmann.ac.il/~dinuri/focs16/CFP.html/

Workshop on Quantum Simulation and Quantum Walks
Czech Technical University
Prague, Czech Republic
November 17-20, 2016
http://wqsqw2016.phys.cz/

QCIT 2016
Washington D.C.
December 8, 2016
http://www-mobile.ecs.soton.ac.uk/events/GC_16_QCIT_CFP.htm/

FSTTCS 2016
Chennai Mathematical Institute
Chennai, India
December 13-15, 2016
http://www.fsttcs.org/

CONTESTS & EVENTS

Microsoft Quantum Challenge
Microsoft recently announced the winners of its "Quantum Challenge," which invited students to use their quantum circuits and quantum noise simulator. The winner, Ph.D. student Thien Nguyen from Australia, was awarded the Grand Prize of $5,000 for his entry "Simulating Dynamical Input-Output Quantum Systems with LIQUi|>". Visit the QuARC research group's web page to learn more about upcoming challenges.
https://aka.ms/quantumchallenge/

ProtoHack
ProtoHack is a code-free hackathon aimed at non-technical entrepreneurs. During the event, participants use wireframing and prototyping tools to design and demo an idea, concluding with a pitch for the other participants and judges, including investors. ProtoHack events will take place in San Francisco (Oct. 15), New York City (Nov. 5), and Calgary (Nov. 26)
http://protohack.org/
https://hackthenorth.com/

Internationalization and Unicode Conference 40
The 40th Internationalization and Unicode Conference will take place November 1-3 in Santa Clara, CA later this year. Unicode is a computing industry standard for the representation and handling of text, and contains more than 128,000 characters. The conference will discuss updates on the latest standards and techniques for troubleshooting common problems or requirements.
http://unicodeconference.org/

GRANTS AND SCHOLARSHIPS

Richard E. Merwin Student Scholarship
Website: *https://www.computer.org/web/students/merwin*
Deadline: *September 30, 2016*
Eligibility: Graduate or undergraduate students currently enrolled in a computer related field who hold a GPA of no less than 2.5 and are members of IEEE Computer Society.
Benefits: $1,000 (for one academic year)
Explanation: The scholarship is sponsored by IEEE to reward active student volunteer leaders participating in student branches or chapters. Winners will serve as IEEE Computer Society Student Ambassadors.

Women in Technology Scholarship
Website: *https://www.buildium.com/women-in-technology-scholarship/*
Deadline: October 1, 2016
Eligibility: Female graduate or undergraduate students currently enrolled in product design, interaction design, UX design, or computer science programs.
Benefits: $2,500
Explanation: Boston-based startup, Buildium, is offerring a scholarship that aims to recognize female STEM leaders. Applicants are required to submit an essay (roughly 1,000 words) describing a female STEM leader who inspires them.

Build U. Scholarship
Website: *https://www.buildium.com/buildiums-build-u-scholarship/*
Deadline: October 1, 2016
Eligibility: Graduate or undergraduate students currently enrolled in product design, interaction design, UX design, or computer science programs.
Benefits: $2,500
Explanation: The scholarship aims to recognize companies that value friendly working environments and customer support. Applicants are required to submit an essay (approximately 1,000 words) describing one company that exemplifies Buildium's core values.

UPE/ACM Scholarship Award
Website: *http://upe.acm.org/scholarship.html*
Deadline: December 15, 2016
Eligibility: Graduate or undergraduate students currently enrolled in a computer-related field, who are student members of the ACM and a member of an ACM Student Chapter at an academic institution.
Benefits: $1,000
Explanation: The scholarship is sponsored by UPE and the ACM to raise the importance of academic achievement and professional commitment for ACM student members looking to enter into the computing profession.

The Generation Google Scholarship
Website: *https://www.google.com/edu/scholarships/the-generation-google-scholarship/*
Deadline: January 2017 (The official application deadline has not been set, make sure to check the website for further details.)
Eligibility: Graduate or undergraduate students who belong to an underrepresented group in computer science at a university in the U.S. or Canada.
Benefits: $10,000
Explanation: The scholarship aims to help aspiring computer scientists excel in technology and become leaders in the field. Winners will be invited to attend the Google Scholars' Retreat in the summer of 2017.

FEATURED EVENT

Quantum Communications and Information Technology (QCIT '16)

Washington D.C.
December 8, 2016

Unlike in the classical public key cryptography, which uses one-way functions, quantum key distribution (QKD) uses the principles of quantum mechanics to generate and share keys between two parties. It also allows the provision to detect eavesdropping during the sharing process. The resultant shared keys can then be used with any chosen encryption algorithms.

Though these concepts have been fairly developed, there is no real system on the market that can carry out these tasks. This conference aims to connect people from academia and industry to exchange ideas related to quantum communications and developing applications and research in this new field.

QCIT '16 is a part of IEEE Globecom, which will be held in the U.S. capital of Washington D.C. Conference goers can visit various points of interest like the Smithsonian Institution, the Washington Monument, the Lincoln Memorial, and the iconic White House.

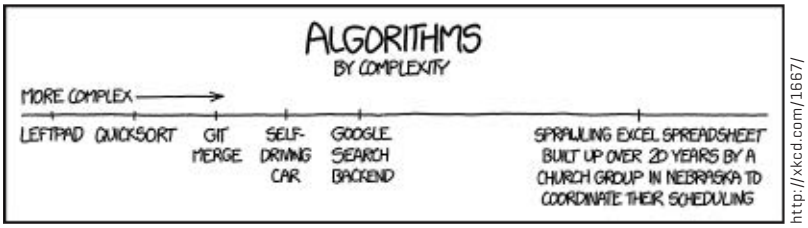
For more information, please visit *http://www-mobile.ecs.soton.ac.uk/events/GC_16_QCIT_CFP.htm/*.

—Darshit Patel

Image by Carol VanDyke

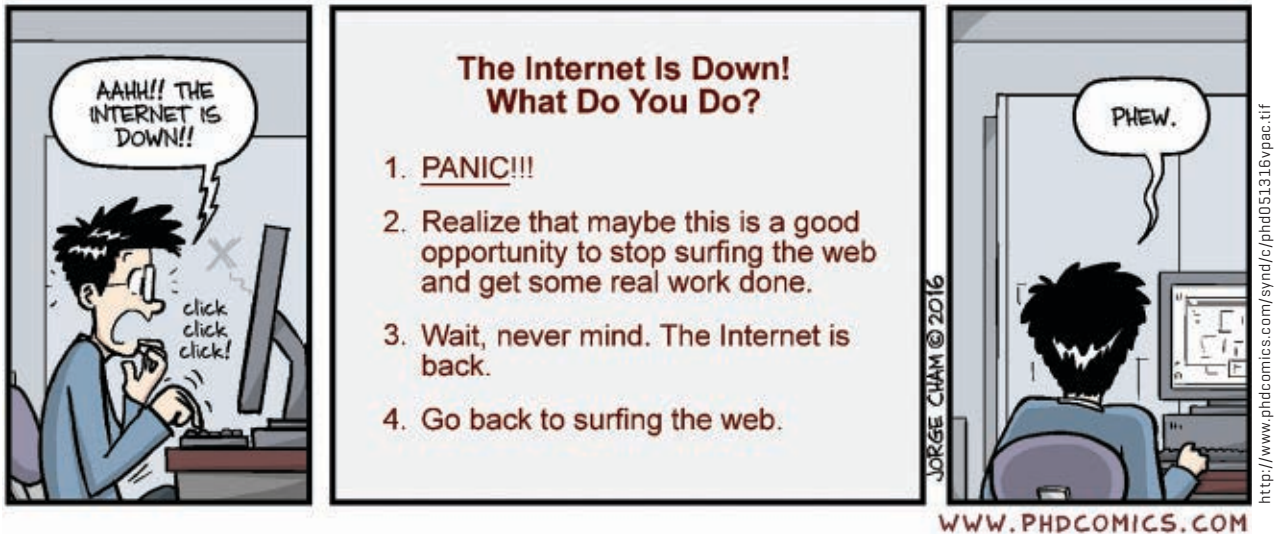
BEMUSEMENT

Algorithms



http://xkcd.com/1667/

Internet Goes Down



http://www.phdcomics.com/synd/c/phd051316vpac.tif

A Question of Time

If the hour and minute hands are at equal distance from the 6 hour, what time will it be exactly?

Find the solution at: http://xrds.acm.org/bemusement/2016.cfm

SUBMIT A PUZZLE

Can you do better? Bemusements would like your puzzles and mathematical games (but not Sudoku). Contact **xrds@acm.org** to submit yours!

XRDS

Seeks Student Volunteers

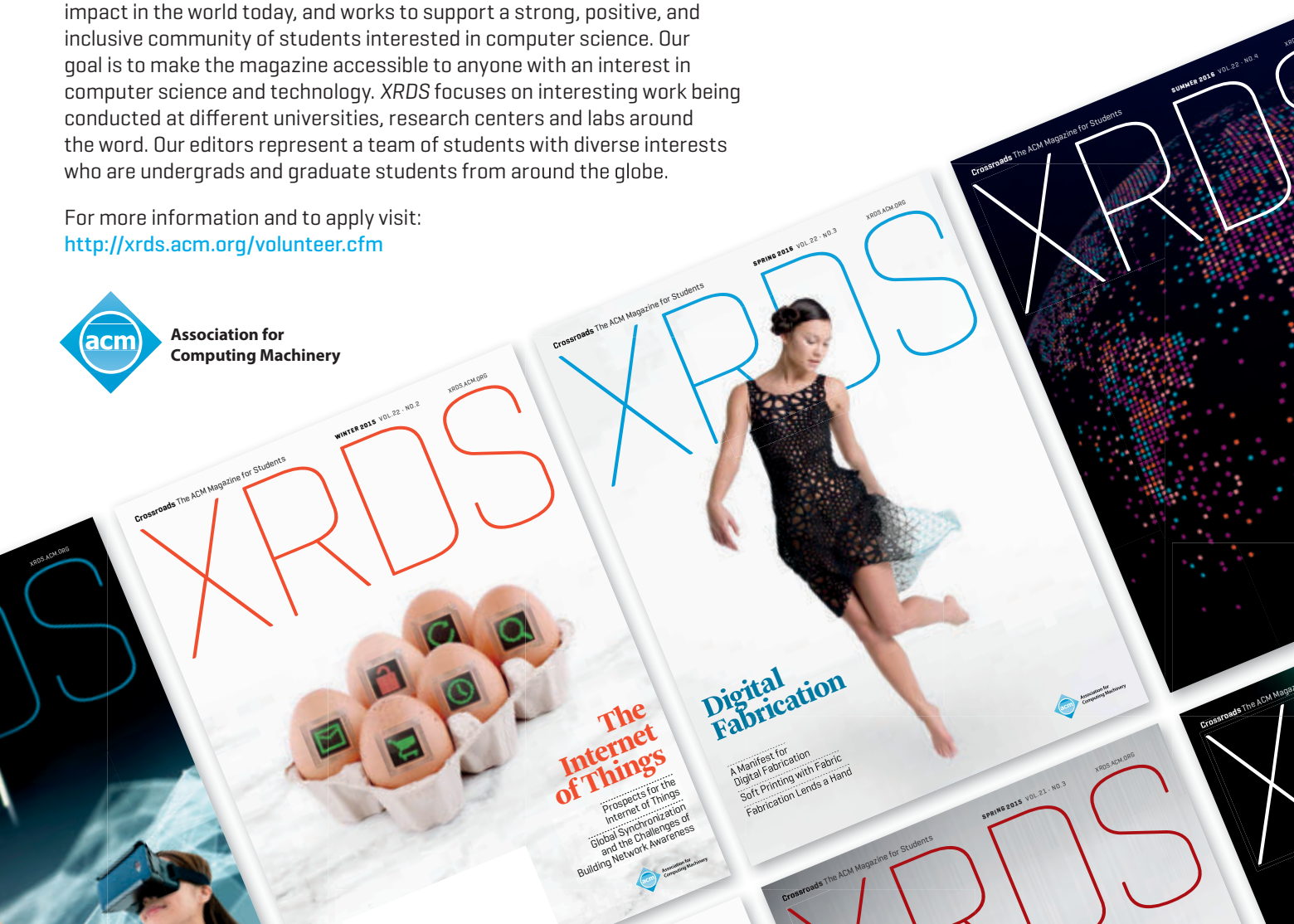
Are you a student who enjoys staying up to date with the latest tech innovations, and is looking to make an impact on the ACM community?

Editorial positions are now open for the following positions: social media editor, feature editor and department editor.

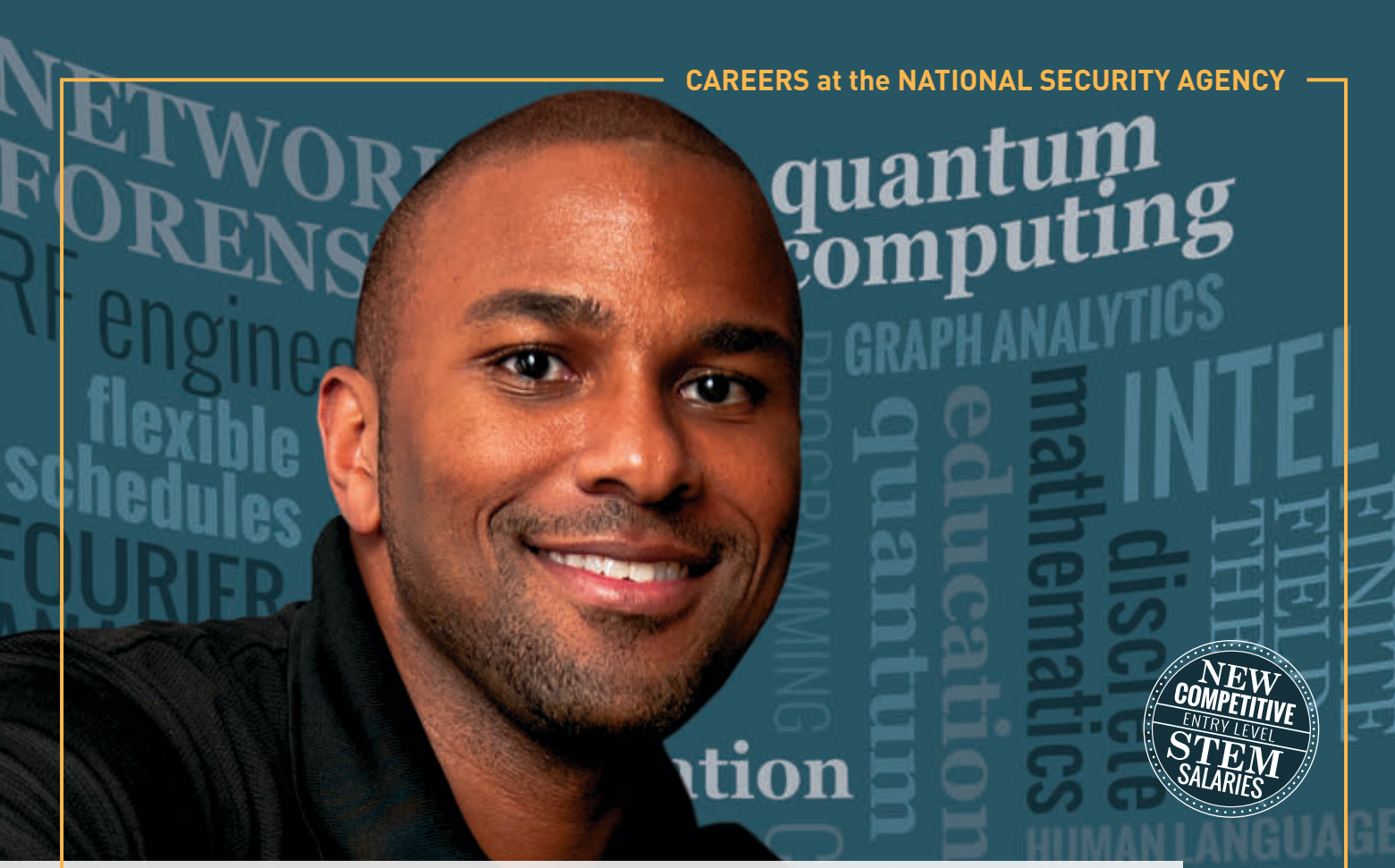
XRDS is a quarterly print magazine for students by students that examines cutting edge research in computer science, viewpoints on technology's impact in the world today, and works to support a strong, positive, and inclusive community of students interested in computer science. Our goal is to make the magazine accessible to anyone with an interest in computer science and technology. *XRDS* focuses on interesting work being conducted at different universities, research centers and labs around the word. Our editors represent a team of students with diverse interests who are undergrads and graduate students from around the globe.

For more information and to apply visit:
http://xrds.acm.org/volunteer.cfm

Association for Computing Machinery



CAREERS at the NATIONAL SECURITY AGENCY



EXTRAORDINARY WORK

Inside our walls, you will find the most extraordinary people doing the most extraordinary work. Yet it's not just finite field theory, quantum computing or RF engineering. It's not just discrete mathematics or graph analytics.

It's all of these and more – rolled up into an organization that leads the world in signals intelligence and information assurance. Inside our walls you will find extraordinary people, doing extraordinary work, for an extraordinary cause:

The safety and security of the United States of America.

U.S. citizenship is required for all applicants. NSA is an Equal Opportunity Employer and abides by applicable employment laws and regulations. All applicants for employment are considered without regard to age, color, disability, genetic information, national origin, race, religion, sex, sexual orientation, marital status, or status as a parent.

- Computer/Electrical Engineering
- Computer Science
- Cybersecurity
- Information Assurance
- Mathematics
- Foreign Language Analysis
- Intelligence Analysis
- Cryptanalysis
- Signals Analysis
- Business
- Finance & Accounting
- Paid Internships, Scholarships & Co-op

APPLY TODAY

NSA

WHERE INTELLIGENCE GOES TO WORK®

Search NSA to Download

Download on the App Store | GET IT ON Google Play

IntelligenceCareers.gov/NSA