Wie rechnen Quanten?

Prof. Dr. Ehrhard Behrends
Fachbereich Mathematik und Informatik
Freie Universität Berlin
Arnimallee 2 – 6
14195 Berlin

behrends@math.fu-berlin.de

Dieser Artikel ist eine gekürzte Version des Artikels gleichen Titels, der in dem Buch Alles Mathematik [1] *erschien und den wir hier mit freundlicher Genehmigung des Vieweg Verlags abdrucken dürfen. Das Buch* Alles Mathematik*, das eine Vielzahl weiterer interessanter populärwissenschaftlicher Artikel über mathematische Anwendungen enthält, erscheint 2008 in der dritten Auflage.*

Die Idee, Gesetze der Quantenmechanik für den Bau von Computern mit neuen Eigenschaften zu bauen, geht auf den Physiker *R. P. Feynman* zurück: 1982 schlug er vor, für die komplizierten Rechnungen im Zusammenhang mit Elementarteilchen-Modellen eigens dafür konzipierte Rechner zu verwenden. Diese noch sehr vage Idee wurde von *D. Deutsch* aufgegriffen, der Ende der achziger Jahre ein theoretisches Modell — also so etwas wie einen möglichen Bauplan — für einen Quantencomputer entwarf.

All das war allerdings nur wenigen Spezialisten bekannt. Gewaltiges Aufsehen erregten erst Arbeiten von *P. Shor*. Der konnte zeigen: Wenn es gelingt, einen funktionierenden Quantencomputer zu bauen, dann sind gewisse Verfahren der Kryptographie, die heute als absolut sicher gelten, relativ schnell zu überlisten.

Peter Shor bekam für diese Leistung 1998 auf dem Weltkongress der Mathematiker in Berlin den Nevanlinna-Preis, das ist die mit Abstand höchste Auszeichnung, die man für Arbeiten aus dem Bereich Informatik/Mathematik bekommen kann. Nach allgemeiner Überzeugung hat er diesen Preis auch verdient, denn seit seinen grundlegenden Arbeiten ist wirklich ein neues Kapitel physikalischer Forschung entstanden. Probleme der Quantencomputer, Quantenkryptographie und der Quanteninformation werden heiß diskutiert, im Umfeld gibt es Detailprobleme, die mit viel Geld gefördert werden (Teleportation, Selbstkorrektur von Quantenkanälen, ...).

Trotz aller Anstrengungen gibt es heute noch keinen Quantencomputer, der diesen Namen verdient, und viele meinen sogar, dass das zu unseren Lebzeiten auch nicht passieren wird (wenn überhaupt jemals). Die physikalischen Schwierigkeiten sind immens; sie sollen aber hier nicht diskutiert werden, denn sie können von Fachleuten aus der Physik viel besser dargestellt werden. Ziel des Artikels ist vielmehr ein Teilaspekt des Themas „Quantencomputer“: Welche neue *Mathematik* wird benötigt, um ihre neuen Möglichkeiten voll auszuschöpfen? Es soll versucht werden zu erklären, wie denn ein Angriff à la Shor auf ein sicheres Kryptosystem aussehen könnte; jedenfalls wenn es die Physiker irgendwann einmal schaffen würden, die Schwierigkeiten zu überwinden.

Warum sind Primzahlen in der Kryptographie wichtig?

Im Artikel von M. Meiringer über Kryptographie auf S. 48 wurde gezeigt, dass die Sicherheit des berühmten *RSA-Algorithmus* auf der Schwierigkeit der Faktorisierung großer Zahlen $n = p \cdot q$ beruht. Es gibt zur Zeit kein Verfahren, das wesentlich besser wäre als systematisches Probieren, um aus n auf p und q zu schließen.

> Als ein für die Kryptographie realistisches Beispiel gehen wir davon aus, dass p und q jeweils 200 Stellen haben. Dann hat $n = p \cdot q$ vierhundert Stellen, und Ausprobieren erfordert 10^{200} (eine 1 mit zweihundert Nullen!) Rechenschritte. Für derartige Zahlen gibt es schon keine eigenen Namen mehr. Man kann sich überlegen, dass alle Computer dieser Welt an diesem Faktorisierungsproblem scheitern müssen. Selbst wenn sie hundertmal so schnell rechnen wie heute theoretisch möglich ist. Und selbst wenn sie sich seit Beginn der Welt nur diesem Problem gewidmet hätten.

Allerdings ist das RSA-Verfahren nur so lange sicher, wie niemand in der Lage ist, p und q aus n herauszulesen. Denn wer das könnte, wäre auch in der Lage, die verschlüsselte Nachricht zu decodieren. Oben wurde ausgeführt, warum man heute meint, dass RSA sicher ist. Aber:

Quantencomputer könnten schnell p und q ermitteln!

Dafür hat Peter Shor 1994 ein Verfahren vorgestellt. Trotzdem können alle Kryptographen noch relativ ruhig schlafen, weil funktionierende Quantencomputer in weiter Ferne sind. Die zugrunde liegenden mathematischen Ideen sind aber interessant, und um die soll es im Folgenden gehen.

Eine mathematische Vorbereitung: Periodenlängen

Das Problem besteht darin, die zwei Faktoren einer aus zwei Primzahlen zusammengesetzten Zahl zu finden: Rekonstruiere p, q aus $n := p \cdot q$. Viele Verfahren sind erdacht worden, um das zu erleichtern oder umzuformen. Für unsere Zwecke ist eine Idee wichtig, die das Faktorisierungsproblem in eine andere Fragestellung transformiert, die für klassische Computer haargenau den gleichen Schwierigkeitsgrad hat.

Um sie vorzustellen, muss eine Vokabel eingeführt werden: Was heißt „a *modulo* b“? Es sollen zwei Zahlen a und b gegeben sein, in der Regel ist a viel größer als b. Zunächst teilt man a durch b und schaut sich den Rest an, der beim Teilen übrig bleibt. Diese Zahl, sie liegt zwischen 0 und $b-1$, wird „a modulo b“ genannt. Modulares Rechnen wurde auch im Kryptographie-Artikel (vgl. S. 48) betrachtet.

Dieses „modulo“ spielt nun eine wichtige Rolle. Wir beginnen mit einer Zahl n, die die Form $p \cdot q$ mit Primzahlen p, q hat, als illustrierendes Beispiel denken wir an $n = 15 = 3 \cdot 5$. Jemand gibt nun eine Zahl x vor, die irgendwo zwischen 1 und n liegt. Kann man x dazu verwenden, einen Faktor von n zu finden?

Ideal wäre es, wenn n und x einen von 1 verschiedenen Teiler gemeinsam hätten (wenn also in unserem Beispiel etwa $x = 10$ wäre). Jeder Computer findet den in Bruchteilen von Sekunden, das Verfahren heißt „Euklidischer Algorithmus“ und war schon vor über 2000 Jahren bekannt.

Wir nehmen also an, dass x und n *keinen* Teiler gemeinsam haben, man sagt, dass x und n *teilerfremd* sind. Dann rechnen wir nach und nach die Zahlen

x modulo n,

x^2 modulo n,

x^3 modulo n,

...

aus. Die Zahlentheorie kann beweisen, dass mit Garantie irgendwann einmal die Zahl 1 herauskommt. Wir nennen denjenigen Exponenten r, für den zum ersten Mal x^r modulo n gleich 1 ist, die *Periode* von x.

> In unserem Beispiel $n = 15$ starten wir zur Illustration mit $x = 7$. Die Zahlen n und x sind teilerfremd, wir können also die Periode von 7 ausrechnen.
>
> Dazu müssen wir so lange die Reste von $7, 7^2, 7^3, \ldots$ modulo 15 bestimmen, bis wir erstmals 1 erhalten:
>
> 7 modulo 15 ist gleich 7;
>
> 7^2 modulo 15 ist gleich 4;
>
> 7^3 modulo 15 ist gleich 13;
>
> 7^4 modulo 15 ist gleich 1.
>
> Folglich ist die Periode von $x = 7$ gleich 4.

Was nutzt das? Angenommen, unser x ist so, dass die Periode r eine gerade Zahl ist: $r = 2 \cdot s$. Dann können wir doch die Gleichung[8] $x^r = 1$ unter Verwendung der Abkürzung $y := x^s$ als $y^2 = 1$ bzw. als $(y+1) \cdot (y-1) = 0$ umschreiben. Und „gleich 0 modulo n“ bedeutet Teilbarkeit durch n und damit durch p und q. Wenn man nun noch die Tatsache verwendet, dass eine Primzahl ein Produkt nur dann teilt, wenn es einen der Faktoren teilt, so liefert uns die Kenntnis von y die Kenntnis von p und q. (Ich habe ein bisschen geschummelt: Für die Argumentation ist wichtig, dass $y+1$ nicht Null modulo n ist. Das erklärt den Zusatz in der nachstehenden Definition).

Zusammengefasst können wir also sagen, dass wir aus der Periode von x „mit etwas Glück“ einen Teiler von n bekommen. Wir präzisieren das in der

Definition: Eine Zahl x zwischen 1 und n soll *gut* heißen, wenn x zu n teilerfremd ist, die Periode von x eine gerade Zahl $r = 2 \cdot s$ ist und x^s modulo n *nicht* die Zahl $n-1$ ist.

Bemerkenswerterweise ist es nun so, dass es gute Zahlen im Überfluss gibt[9]. Greift man zufällig eine heraus, so ist sie mit mehr als fünfzig Prozent Wahrscheinlichkeit gut. Das führt zur folgenden **Strategie zur Lösung des Faktorisierungsproblems:**

- Suche mit einem Zufallsgenerator eine Zahl x zwischen 1 und n.
- Mit sehr viel Glück hat sie einen gemeinsamen Teiler mit n, dann ist man fertig. Mit mindestens fünfzig Prozent Wahrscheinlichkeit ist x gut, und dann kann man mit Hilfe der Kenntnis der Periode ebenfalls faktorisieren.
- Sollte man kein Glück gehabt haben, wiederhole man die ersten beiden Schritte. Irgendwann wird es schon klappen, denn laut Wahrscheinlichkeitsrechnung ist die Wahrscheinlichkeit für „Pech“ in k aufeinanderfolgenden Schritten höchstens $0{,}5^k$. Sie müssen schon ein echter Pechvogel sein, wenn es zehnmal schiefgeht, diese Wahrscheinlichkeit ist kleiner als ein Promille.

Für sich genommen ist das eine interessante, aber recht nutzlose Umschreibung des Problems, denn:

> **Für einen klassischen Computer ist das Berechnen der Periode genauso kompliziert wie das Faktorisieren selber!**

(Das Finden von Zufallszahlen x dagegen ist leicht, das gehört heutzutage zu den Standardaufgaben.)

Hier sollen auf spektakuläre Weise Quantencomputer zum Einsatz kommen. Ihre einzige Aufgabe (beim Faktorisierungsproblem) besteht darin, für ein vorgelegtes x die Periode von x zu bestimmen. Alles andere, also das zufällige Erzeugen von x und die weiteren Rechnungen wie etwa die mehrfache Ausführung des euklidischen Algorithmus, kann den klassischen Rechnern überlassen bleiben.

[8] Wir lassen das „modulo“ der Einfachheit halber weg und rechnen so wie mit gewöhnlichen Zahlen; das ist wirklich legitim!

[9] Der Beweis ist elementar, aber etwas länglich.

Etwas Quantenmechanik

Hier wollen wir uns wirklich auf das Notwendigste beschränken. Auch nach 100 Jahren ist die Quantenmechanik immer noch eine Wissenschaft, die den meisten ein Buch mit sieben Siegeln ist. Das ist auch ganz verständlich, denn es handelt sich um ein (hervorragend funktionierendes) Modell der Welt im atomaren Bereich, das nach Gesetzen funktioniert, die der menschlichen Lebenserfahrung total zuwiderlaufen.

Was uns interessiert, kann am folgenden Beispiel demonstriert werden. Wir denken an eine Situation, bei der Teilchen von atomarer Größenordnung betrachtet werden und bei der durch die Versuchsanordnung klar ist, dass genau eine von zwei Möglichkeiten verwirklicht werden kann:

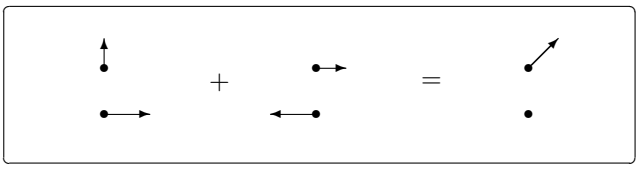
- Ein Photon, das auf eine Glasplatte schräg auftrifft, kann hindurchgehen oder gebrochen werden;
- ein Elektron kann bei einer Messung einen Spin „up“ oder „down“ haben;
- ein Teilchen (etwa ein zu einem Atomkern gehörendes Elektron) kann genau eines von zwei Energieniveaus einnehmen.

Wir wollen für den Augenblick die möglichen Ergebnisse A und B nennen. Fundamental ist dann die Feststellung, dass das Weitestgehende, das sich theoretisch aussagen lässt, *wahrscheinlichkeitstheoretische Aussagen* sind: Physiker können eine Zahl a zwischen 0 und 1 berechnen, so dass die Wahrscheinlicheit für eine A-Messung gleich a (und folglich die für eine B-Messung gleich $1-a$) ist. Es ist also — etwas unwissenschaftlich ausgedrückt — so, als ob ein Photon kurz vor Erreichen der Glasplatte würfelt, ob es nun hindurchgehen will oder lieber reflektiert werden möchte. Auf die philosophischen Probleme in diesem Zusammenhang können wir hier natürlich nicht eingehen.

Die Wahrheit ist etwas komplizierter, leider benötigen wir gleich diese Verfeinerung. Sie besagt:

> Man stelle sich A und B als Punkte der Ebene vor. An A und B denke man sich „Pfeile“ angebracht, die eine beliebige Richtung haben können. Einzige Bedingung: Misst man die Längen l_A und l_B dieser Pfeile, so muss die Zahl $l_A^2 + l_B^2$ den Wert 1 ergeben.
>
> Die Wahrscheinlichkeit, A zu messen, ist dann gerade die Zahl l_A^2.

Zwei Ergänzungen sind nun noch wichtig. *Erstens* ist es so, dass dieses Pfeil-Bild den Zustand *vor* der Messung beschreibt. Wird zum Beispiel gemessen, dass A eingetreten ist, so verändert sich der Zustand schlagartig: Es gibt nun einen Pfeil der Länge 1 bei A, und der bei B ist verschwunden. Aus dem Blickwinkel der Quantencomputer ist es bedauerlicherweise so, dass „Messung“ sehr weit interpretiert werden muss, jede Wechselwirkung mit anderen Systemen hat die gleiche Auswirkung wie eine Messung. Und *zweitens* kann es so etwas wie eine *Überlagerung* geben. Das muss man sich so vorstellen, dass man manchmal nicht weiß, welches von zwei A-B-Pfeildiagrammen für die Beschreibung einer Situation das richtige ist, zum Beispiel, weil ein Photon sich für einen von zwei Spalten zum Durchgehen entschieden hat, wir aber nicht wissen, für welchen. Dann kommt es zur Überlagerung, das richtige Modell entsteht dann so, dass man die Pfeile der einzelnen Modelle per Vektoraddition zusammensetzt (und hinterher den Maßstab noch so abändert, dass die Summe der Längenquadrate wieder Eins ist). Da sich Vektoren je nach Richtung verstärken, abschwächen oder sogar ganz auslöschen können, kommt es zu merkwürdigen Phänomenen, die klassisch nicht erklärbar sind.



Qbits: Die Bausteine eines Quantencomputers

Der Ausgangspunkt ist ganz einfach, wir knüpfen an den vorigen Abschnitt an: Grundbaustein eines Quantencomputers ist eine physikalische Situation, die bei Messung genau eines von zwei Ergebnissen produziert. Eben noch haben wir sie A und B genannt, ab jetzt sollen sie 0 und 1 heißen. Man spricht dann von einem **Qbit**. (Der Name soll natürlich daran erinnern, dass der Grundbaustein eines klassischen Computers ein Bit ist, also eine Einheit, die die Werte 0 und 1 annehmen kann.)

Der wesentliche Unterschied ist der folgende: Ein **Bit** ist in einem der Zustände 0 oder 1. Definitiv. Ein **Qbit** dagegen ist mit einer gewissen Wahrscheinlichkeit in 0 bzw. 1, die einzelnen Wahrscheinlichkeiten werden durch die Länge der Pfeile bei 0 bzw. 1 bestimmt. Nur durch eine Messung können wir den Zustand erfahren, der dann aber unwiederbringlich verändert wurde. Rein formal gesehen ist ein Bit ein spezielles Qbit, ein Bit im Zustand 0 etwa entspräche einem Qbit, bei dem 0 einen Pfeil der Länge Eins trägt (und der Pfeil bei 1 verschwindet).

Nun kann man mit einem Qbit recht wenig anfangen, wir brauchen viele. Die Lösung kann nicht darin bestehen, einfach einzelne Qbits nebeneinanderzupacken, man möchte auch noch die Wechselwirkungen ausnutzen.

Nehmen wir etwa zwei Qbits, Q1 und Q2. Sind beide im Zustand 0, so wollen wir den Gesamtzustand mit 00 bezeichnen, analog sind die Zustände 01, 10

und 11 zu verstehen. Überlassen wir beide sich selber, so wird irgendeiner dieser gemeinsamen Zustände vorliegen, wir wissen aber nicht, welcher. Wieder ist es so, dass nur Wahrscheinlichkeiten vorausgesagt werden können. Diesmal sind vier Pfeile — je einer für $00, 01, 10, 11$ — vorzuschreiben. Die Längen, zum Quadrat genommen, müssen sich zu Eins summieren, diese Quadrate stehen für Wahrscheinlichkeiten. Hat etwa der Pfeil bei 01 die Länge $0{,}7$, so werden wir mit Wahrscheinlichkeit $0{,}7 \cdot 0{,}7 = 0{,}49$ (das sind 49 Prozent) Q1 im Zustand 0 und Q2 im Zustand 1 messen.

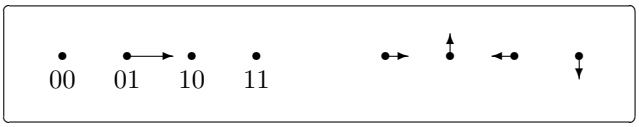
Mit wachsender Anzahl der Qbits sind nun immer mehr Pfeile zu beherrschen. Genauer: Bei L Qbits spielen 2^L Pfeile eine Rolle. Es ist dieser Punkt, der zu den gewaltigen Möglichkeiten führt, man könnte es als massive Parallelität des Rechners interpretieren[10].

Nach diesen Vorbereitungen können wir das **Anforderungsprofil** für einen Quantencomputer grob spezifizieren.

Forderung 1: Der Quantencomputer muss über L Qbits verfügen. Für die Anwendung, die wir planen, ist $L = 2.000$ eine realistische Größenordnung. Es muss möglich sein, diese Qbits beliebig zu initialisieren: Denken wir uns beliebige 2^L Pfeile aus, für die die quadrierten Längen sich zu Eins summieren, so muss es möglich sein, den Computer in einen Zustand zu bringen, dass die 2^L möglichen Zustände durch genau diese Pfeile charakterisiert sind.

Ein Computer, in denen man nur etwas einlesen kann, ist noch ziemlich uninteressant, daher folgt nun die

Forderung 2: Es muss möglich sein, an diesen Qbits im Wesentlichen die gleichen Operationen vorzunehmen wie an gewöhnlichen Bits: Negation, Konjunktion, Disjunktion, … (Das ist ein heikler Punkt, denn man kann aus theorie-inhärenten Gründen einen Quantenzustand nicht kopieren, ohne vorher eine Messung vorzunehmen, durch die die vorher mühsam produzierte Überlagerung der einzelnen Zustände schlagartig aufgehoben würde.) Es kommt noch eine spezielle Forderung hinzu, die für unsere Zwecke unerlässlich ist. Der Quantencomputer muss die diskrete *Fouriertransformation* beherrschen. Vereinfacht ausgedrückt geht es darum, mittels einer rechner-internen Operation — die die Überlagerungen nicht zerstört — einen bei einem Zustand stehenden Pfeil in eine Pfeilfamilie zu verwandeln. Dabei wird der Ausgangspfeil hergenommen, stark verkürzt und unter gewissen Drehwinkeln an die einzelnen Zustände angehängt. Nachstehend sehen wir einen Ausgangszstand (er ist deterministisch beim Zustand 01) und daneben seine diskrete Fouriertransformation; oberflächlich gesehen wird zunächst garantiert Zustand 01 bei einer möglichen Messung produziert, und nach Transformation sind alle Zustände gleichwahrscheinlich.



Das Wichtigste aber ist, dass wir auch noch die Richtungen der Pfeile kontrollieren können, das wird gleich entscheidend sein.

Als letztes bestehen wir noch auf

Forderung 3: Es muss möglich sein, die einzelnen Operationen so durchzuführen, dass Überlagerung gewährleistet ist, dass sich also die Wahrscheinlichkeitspfeile entsprechend der Vektoraddition zusammensetzen. Sind also zum Beispiel zu Beginn nur zwei Zustände Z1 und Z2 — beschrieben durch die Pfeile P1 (bei Z1) und P2 (bei Z2) — möglich und werden mit Z1 und Z2 Rechnungen durchgeführt, die zu Ergebnissen E1 und E2 führen, so soll der Rechner nach der Rechnung in einem Gesamtzustand sein, für den die Pfeil-Längen und -Richtungen an den einzelnen möglichen Zuständen aus P1 und P2 sowie aus den zu E1, E2 gehörigen Pfeilen mittels Vektoraddition entstanden sind.

Wie faktorisiert man mit einem Quantencomputer große Zahlen?

Nun können wir die Idee von Shor in den Grundzügen nachvollziehen. Wir erinnern daran, dass wir für eine Zahl $n = p \cdot q$ allein aus dem n die Zahlen p und q finden wollen und dass es reicht, zu einem x zwischen 1 und n die Periode zu berechnen. Das geht nach Shor so:

1. **Schritt:** Die Zahl n ist gegeben, sie soll mit L Ziffern im Zweiersystem darstellbar sein (hat n im Zehnersystem zum Beispiel 90 Stellen, so führt das — da 10^3 ungefähr 2^{10} ist — auf etwa 300 Stellen im Dualsystem). Verschaffe Dir einen Quantencomputer mit $3L$ Qbits. Er soll die im vorigen Abschnitt beschriebenen Eigenschaften haben.
2. **Schritt:** Organisiere den Computer so: Die ersten $2L$ Qbits sollen „das erste Register“ heißen, die letzten L Qbits taufe man als „das zweite Register“.
3. **Schritt:** Suche ein zufälliges x zwischen 1 und n. Das können wir einem klassischen Computer übertragen.
4. **Schritt:** Präpariere den Computer: Zu allen Zuständen der Form $x \cdots x0 \cdots 0$ (irgendwelche Nullen und Einsen in den ersten $2L$ Qbits, nur Nullen im zweiten Register) gehört ein nach rechts zeigender Pfeil, und alle haben die gleiche Länge.

[10] In gewisser Weise führen L Qbits zu einem Computer, der 2^L Zahlen gleichzeitig verarbeiten kann, allerdings wird jede einzelne nur mit einer gewissen Wahrscheinlichkeit eine Rolle spielen. Dieses exponentielle Ansteigen von 2^L mit wachsendem L ist kaum vorstellbar, man kommt sonst damit selten in Berührung. Höchstens einmal bei Kettenbriefen oder beim Wundern über die Fabel vom Schachbrett und den Reiskörnern.

5. Schritt: Verändere den Zustand des Computers auf folgende Weise. Zur Zeit sind doch alle Zustände der Form $a0\cdots0$ gleichwahrscheinlich, wo a den Zustand des ersten Registers bezeichnet. Das soll in einen Gesamtzustand übergehen, bei dem die Wahrscheinlichkeit für $a0\cdots0$ auf ay übertragen wird, wobei y — ein Wert im zweiten Register — für „x^a modulo n, geschrieben im Zweiersystem“ steht. Das ist sinnvoll, da mit n auch y höchstens L Dualziffern hat.

***Zusammen*:** Der Computer ist jetzt in einem Zustand, wo man bei einer Messung im ersten Register ein a und im zweiten das zugehörige x^a modulo n finden würde. Dabei kommen alle möglichen a mit gleicher Wahrscheinlichkeit vor, und immer noch zeigen alle Wahrscheinlichkeitspfeile nach rechts.

6. Schritt: Das ist wohl der entscheidende Schritt. Jetzt müssen Fouriertransformation und Überlagerung gleichzeitig ablaufen, ohne sich zu stören. Nur so kann der gewünschte Effekt erzielt werden, der auf der folgenden Idee beruht.

Erstens entstehen doch bei der Fouriertransformation Wahrscheinlichkeitspfeile, die — je nach Zustand — in alle möglichen Richtungen zeigen. Zweitens werden sie nach dem Gesetz des Kräfteparallelogramms überlagert. Und drittens gilt doch: Zeigen „sehr viele“ Pfeile in verschiedene Richtungen, so ist der resultierende Pfeil sehr klein[11]. Nur dann, wenn die Pfeile alle in die gleiche Richtung zeigen, gibt es eine bemerkenswerte Resultierende.

Im vorliegenden Fall werden nun die Zustände im ersten Register einer Fouriertransformation unterworfen. Es ist dann so, dass nur solche Zustände by — mit einem b der Länge $2L$ aus dem ersten Register und einem y der Länge L aus dem zweiten — einen von Null verschiedenen Wahrscheinlichkeitspfeil haben, wenn $r\cdot b$ ein Vielfaches von 2^{2L} ist. Dabei steht r für die gesuchte Periode von x.

7. Schritt: Nun soll das erste Register gemessen werden (die Werte des zweiten sind nicht so wichtig). Aufgrund der zum vorigen Schritt gemachten Bemerkungen erhalten wir ein b, so dass rb ziemlich genau ein Vielfaches von 2^{2L} sein muss. Mit elementaren Methoden ist es dann leicht, daraus das r zu ermitteln (Kettenbruchentwicklung!).

8. Schritt: Teste, ob x gut ist. Das kann wieder ein klassischer Computer übernehmen, da die Periode bekannt ist. Falls ja, ist damit ein Teiler von n gefunden, die Begründung findet sich oben in Abschnitt 2. Falls nein, fange noch einmal beim dritten Schritt an. Es ist dann ziemlich sicher, dass vergleichsweise schnell die Faktorisierung gefunden wird.

Zusammenfassung

Alles war doch ziemlich verwickelt, daher soll hier noch einmal auf die wichtigsten Punkte hingewiesen werden:

- Gewisse heute als sicher geltende kryptographische Verfahren, insbesondere der public-key-Algorithmus RSA, sind dann nicht mehr sicher, wenn man eine Technik kennt, aus einer zusammengesetzten Zahl $n = p\cdot q$ die Faktoren herauszulesen.
- Wenn es jemand schafft, auf schnelle Weise die Periode eines beliebigen x auszurechenen, so ist das Problem gelöst. Als neuer Aspekt kommt hinzu: Man muss es eventuell mehrfach versuchen, denn das Verfahren ist nur mit einer gewissen positiven Wahrscheinlichkeit erfolgreich. Diese Wahrscheinlichkeit ist in unserem Fall beruhigend hoch (höher als 50 Prozent).
- Quantencomputer könnten genau das leisten. Sie müssen allerdings genügend kompliziert sein (einige tausend Qbits) und gewisse Forderungen erfüllen. Diese sind beim heutigen Stand der Technik nicht einmal ansatzweise zu verwirklichen. Das Hauptproblem ist, ein kompliziertes quantenmechanisches System so abzuschirmen, dass es keine Dekohärenz (= durch Messung oder Wechselwirkung zustande gekommener Verlust des Überlagerungszustands) gibt.

  Falls so ein Quantencomputer wirklich zur Verfügung steht, ist nichts weiter zu tun, als ihn wie oben beschrieben zu präparieren, die erforderlichen Zustandsänderungen durch geeignete Gatter vorzunehmen und dann das erste Register zu messen. Mit einer hohen Wahrscheinlichkeit führt das zur Periode des vorgelegten x.

Ich persönlich glaube nicht, dass man jemals auf diese Weise Zahlen einer interessanten Größenordnung faktorisieren kann. Das Thema „Quantenkryptographie“ hat aber noch andere Aspekte, die sicher schneller zu greifbaren Ergebnissen führen. Dazu gehört zum Beispiel die abhörsichere Übertragung von Schlüsseln: man benötigt im wesentlichen ein einziges Qbit, es sind schon Testläufe über einige Dutzend Kilometer erfolgreich durchgeführt worden.

Da ist die enthaltene Mathematik aber eher uninteressant. Wichtiger sind die beteiligten physikalischen Phänomene, deswegen wurde in diesem Artikel auch nicht darauf eingegangen.

Literaturverzeichnis

[1] M. Aigner und E. Behrends, *Alles Mathematik*, Vieweg, Braunschweig, Wiesbaden, 3. Auflage, 2008.

[11] Davon kann sich jeder am Beispiel resultierender Kräfte überzeugen: Wenn mehrere Hunde an einer Decke in verschiedene Richtungen zerren, wird sich die kaum von der Stelle bewegen.