E-kursuse

"Kvantarvuti ja kvantkrüptograafia alused"

materjalid

Aine maht 3 EAP

Peeter Saari

Tartu Ülikool
2011

Täiendatud - L okt 22 20:35:50 2011

Kvantarvutite põhimõisted, -elemendid, -tõed

1. Kvantbitti (Qubit) kandva süsteemi omaolekud ja kvantbitt-register

Kvantbiti olekute kirjeldamiseks kõigepealt defineerime kaks 2-komponendilist vektorit $|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ $|1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ kui baasivektorid ehk telgede ühikvektorid 2 dim.-ses ruumis.

Sissejuhatavas loengus demonstreeritud katsetes footonite polarisatsiooniga sai viimase kirjeldamiseks samuti sisse tuua vertikaal- ja horisontaaltelje ühikvektorid footoni levisuunaga ristuvas tasandis. Siiski **ei** tule seda kokkulangevust võtta kui sisulist samasust. Valguskiire elektromagnetvälja polarisatsiooni määrab vektor *E* , mis asub füüsikalises 3-dim.-es ruumis (ehkki kiire risttasandis kui 2-dim.-es ruumis). Äsjadefineeritud 2-komponendilised vektorid moodustavad aga baasi abstraktses nn Hilbert'i ruumis, kus üldiselt võib olla dimensioone kuitahes palju, koordinaadid kompleksarvulised jms. Pealegi, kui kvantbitiks on mingi 2-e energiaseisundiga molekul vms, siis neile seisundeile saab vastavusse seada jällegi need kaks 2-komponendilist baasivektorit Hilberti ruumis, ilma et molekulil tarvitseks olla füüsikalises ruumis defineeritud mingi vektoriaalne suurus mingis tasandis.

Baasivektori tähistustes on püstkriips ja nurksulud võetud kvantmehaanikast, nende vahele kirjutatud "1" või "0" aga arvutiteadusest kui biti kaks võimalikku väärtust.

Kummalised sümbolid |0> ja |1> on mugavas nn Dirac'i *bra-ket* tähistuses baasivektorid Hilberti ruumis, kus sümbol 0 ja üks nende sees on ja võib olla suvaline. Näiteks footoni polarisatsiooni kirjeldamise puhul ka näit. püstkriips ja rõhtkriips, tähed V ja H vms. Siinkohal oleme valinud biti kaks väärtust, kuna soovime kvantbitte panna kandma kahendkoodis arve. Seevastu 1 ja 0 baasivektori **komponentidena** ei ole suvalised sümbolid, vaid reaalarvud 1 ja 0 (kui baasivektorid on esitatud komponentide kaudu iseenda baasis, siis nende komponentide väärtused ei saagi olla muud kui 1 ja 0) .

Rohkem kui ühest sõltumatust kvantbitist koosneva liitsüsteemi kui kvantregistri olekuvektor on kvantmehaanika järgi alamsüsteemide olekuvektorite otsekorrutis. Defineerime püstveerg-vektorite otsekorrutise *MC* maatrikstehete abil

☞

Täiendatud - L okt 22 20:35:50 2011

Otsekorutistena avalduvad ka kahe kvantbiti süsteemi olekute baasivektorid, mida tähistame:

$|00\rangle := O(|0\rangle,|0\rangle)$, $|01\rangle := O(|0\rangle,|1\rangle)$, $|10\rangle := O(|1\rangle,|0\rangle)$, $|11\rangle := O(|1\rangle,|1\rangle)$.

Siis

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Seega 2-kvantbitises registris on neli omaolekut, mis võib vastavusse seada kümnendarvudega 0...3:

"0" "1" "2" "3"

Kasutades kümnendtähistust kvantregistri olekuvektori *ket*-sümbolis, näitame otseselt registris hoitavat kümnendarvu, näiteks:

$|6\rangle := O(|1\rangle, O(|1\rangle, |0\rangle))$ $\quad |6\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$

Veendu, moodustades kuni arvu 15-ni vastavaid registriolekuid, et H.ruumi dimensionaalsus.-s kasvab eksponentsiaalselt (alusel 2) registri pikkusega!

$2^3 = 8$

$|5\rangle := O(|1\rangle, O(|0\rangle, |1\rangle))$ $\quad |5\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$

$\langle 5 := \overline{(|5\rangle^T)}$

Defineerides Dirac'i sümboolikas bra-vektori (püstveeru Transponeerimisega horisontaalreaks ja kaaskompleksi võtmisega (katusjoon) see antud juhul küll komponentide reaalsuse tõttu liigne), saame skalaarkorrutise kirjutada kui bra-ket korrutise ja vektorite ortogonaalsus ning normeeritus 1-le avalduvad:

$\langle 5 \cdot |6\rangle = 0$ $\qquad \langle 5 \cdot |5\rangle = 1$

vt. ka --->

2. Superpositsiooniolek (ühtlase amplituudijaotusega)

$|/\rangle := \frac{1}{\sqrt{2}} \cdot (|0\rangle + |1\rangle)$ $\quad |/\rangle = \begin{pmatrix} 0.707 \\ 0.707 \end{pmatrix}$

Defineerime alternatiivsete mõõtmistulemuste saamise tõenäosuste tabeli: (vt ääre taga) --->

$\text{MõõtmistulemusteTNtabel}(|/\rangle) = (0.5 \quad 0.5)$

$|0\text{kuni}15\text{korraga}\rangle := O(|/\rangle, O(|/\rangle, O(|/\rangle, |/\rangle)))$

Pane avaldisele all tema väärtuste näitamiseks võrdusmärk ja interpreteeri vektori |0kuni15korraga> komponentide arvväärtusi, andes mitmesuguseid oma- ja superpos.-olekute |0>, |1> ja |/> kombinatsioone tema definitsioonis (O-avaldis ülal) !

|0kuni15korraga>

Pane siia avaldisele tema väärtuste näitamiseks võrdusmärk ja võrdle vektori komponente (amplituude) tõenäosustega!

MõõtmistulemusteTNtabel(|0kuni15korraga>)

$$\sum \text{MõõtmistulemusteTNtabel}(|0\text{kuni15korraga}\rangle) = 1$$

Millega põhjendub ülalseisev võrdus 1-ga?

3. Kvantbiti suvaline olek ja selle kujutamine Bloch'i vektori abil

Meenutades loengul tehtud katseid laserikiire polarisatsioonist ja 2-D Hilberti ruumi sissetoomist tavalise klassikalise (kuid 1-le kvandile/footonile normeeritud) tasalaine polarisatsioonioleku kirjeldamise eeskujul, leiame, et footoni polarisatsiooniolek ehk üldiselt suvalise 2-e omaolekuga kvantsüsteemi olek on määratud parajasti kahe reaalarvulise nurkparameetriga kujul

$$|\alpha,\phi\rangle := \cos(\alpha)\cdot\mathbf{e_1} + \sin(\alpha)\cdot\exp(i\cdot\phi)\cdot\mathbf{e_2}$$

, kus **e**-d on laine/footoni levisuunaga ja omavahel risti olevad ühikvektorid ehk, üldiselt, Hilberti ruumi baasivektorid

Komponentides sama:

$$|\alpha,\phi\rangle := \cos(\alpha)\cdot\begin{pmatrix}1\\0\end{pmatrix} + \sin(\alpha)\cdot\exp(i\cdot\phi)\cdot\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}\cos(\alpha)\\ \sin(\alpha)\cdot\exp(i\cdot\phi)\end{pmatrix}$$

Nurk alfa näitab elektriväljavektori peamise võnkesuuna kallet koordinaattelgede suhtes ja nurk fii annab (silmas pidades laine ajalist sõltuvust ja kasutades võnkefaaside kompleksarvulist mugavat kirjeldust) selle vektori tipu trajektoori elliptilisust. Selline nurkade abil kirjaviis tagab automaatselt, et olekuvektor Hilberti ruumis on ühikulise pikkusega. Ka esimesele komponendile mingi faasinurgateguri juurdekirjutamine (st vabade parameetrite arvu tõstmine 3.-le) ei ole tarvilik, sest nii elektriväljavektori kui ka kvantmehaanilise olekuvektori puhul on oluline vaid komponentide vaheline suhteline faas.
Seega, kui keerata esimene koordinaattelg vertikaalselt püsti ja 2-te ristuvat horisontaaltelge kasutada teise st kompleksarvulise komponendi kujutamiseks, täidaksid kõikvõimalike olekuvektorite otsapunktid n.ö. põhjapoolkera pinna. Lõunapoolkera pole eespool tehtud märkust faasidest silmas pidades tarvis, sest näiteks lõunapoolusele vastav laine polarisatsioon oleks sama vertikaalne st põhjapooluse-suunalisega identne. Siit ka üks ebameeldivus: ekvaatoripunktidele vastavad polarisatsioonid on samuti kõik identsed -- horisontaalsed, sest kui $\cos(\alpha)$ = 0, siis $\exp(i\cdot\phi)$ on füüsikaliselt sisutu ja võib olla suvaline ilma et polarisatsiooniolek sellest sõltuks. Kokkuvõttes saame kena geomeetrilise kujutamisviisi, kui venitame põhjapoolkera pinna ka lõunapoolkeraks nii, et endine ekvaator läheb lõunapooluseks ja kõik põhjalaiusnurgad kahekordistuvad, st endine alfa = 45 kraadi saab nüüd olema ekvaatoril.

Seega, toome sisse sfääriliste koordinaatide n.-ö. põhjapoolusest loetava laiuskraadi θ ja

asimuutnurga ϕ. Siis vastab igale paarile nende nurkade väärtustele, näiteks:

$$\theta := 30 \cdot \text{deg} \qquad \phi := 55 \cdot \text{deg}$$

üks punkt sfääri pinnal ja selle punkti raadiusvektor kujutab ühte võimalikku olekuvektorit piltlikul, ehkki abstraktsel **Poincare**' sfääril (kui tegemist footonitega) või -- teise nimega -- **Bloch**'i sfääril (kui tegemist elektronide või suvaliste 2-Dim olekuruumiga kvantsüsteemidega, näiteks kahe-energia-tasemelisena vaadeldavate molekulitega).
Poincare' sfäärist piltliku ettekujutuse saamiseks saab allpool proovida nurkade muutmist tema interaktiivsel *MC*-mudelil.

Alljärgneva mudeli joonistamise köök siin peidus - E aug 29 15:05:50 2011

Seega on kvantbiti suvaline olekuvektor avaldatav kujul:.

$$|\psi\rangle := \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) \cdot \exp(i \cdot \phi) \end{pmatrix} \qquad |\psi\rangle = \begin{pmatrix} 0.966 \\ 0.148 + 0.212i \end{pmatrix} \qquad \langle\psi| := \overline{\left(|\psi\rangle^{T}\right)}$$

Normeering õige!

$$\langle\psi| \cdot |\psi\rangle = 1$$

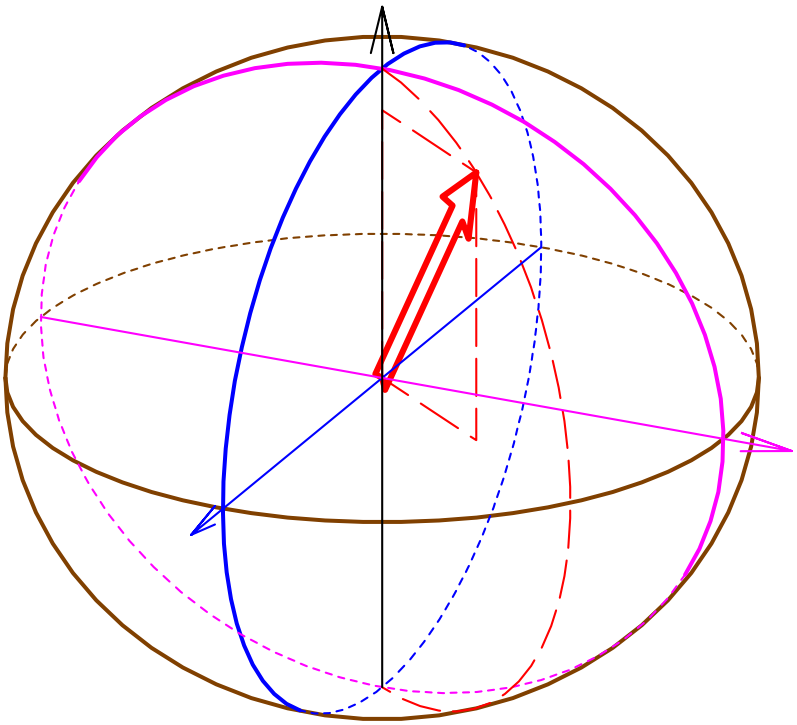
Elektroni või üldse poolespinnilise mikroosakese puhul on Blochi vektoril otseselt geomeetriline sisu tavalises füüsikalises 3-Dim ruumis --- poole Plancki konstandiga korrutatud Blochi vektor annab osakese omapöörlemismomendi (ehk spinni) vektori füüsikalises ruumis, **kuid ainult keskväärtusena** (st üle mõõtmiste identsete objektide ansamblil) , sest spinnvektori komponendid ei oma korraga kindlaid väärtusi, neid mõõtes saame alati juhuslikult kolmikkombinatsioone 2-st diskreetsest omaväärtusest. Näiteks poole Plancki konstandi ühikutes vastab vertikaalsele (z) suunalisele spinnvektori komp.-le operaator σ_z, mille keskväärtus on $\cos(\theta)$ st projektsioon püstteljele, nagu peabki (vt punast kriipsvertikaali pildil allpool)

$$\sigma_z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad \text{keskv}\sigma_z := \langle\psi| \cdot \sigma_z \cdot |\psi\rangle = \cos(\theta) = 0.866$$

see on tuntud kvantmehaanika valem keskväärtuse leidmiseks

$$\text{keskv}\sigma_z = 0.866$$

Bloch'i sfäär



Muuda Blochi sfääri/vektori pildi lähteandmeid $\theta \equiv 30 \cdot \mathrm{deg}$ ja $\phi \equiv 55 \cdot \mathrm{deg}$!
Enne pildi ümberarvuda laskmist püüa ennustada nurkadele antud väärtustele vastav Blochi vektori asend ! (vt. ka vektori komponentide reaal- ja imaginaarosa arvväärtusi)

4. Loogikalülid (Gates)

Kõigepealt käsitleme ühe kvantbitiga opereerivaid lülisid, st skemaatiliselt ühe sisend"juhtmega" ja ühe väljund"juhtmega" "kasti".

Ühe kvantbiti oleku muutus/evolutsioon mingis "raudvaras" (kuid **mitte** "mõõtmise" aktis!) kirjeldub unitaarse operaatoriga/maatriksiga üldkujul:

$$U(\alpha,\phi) := \begin{pmatrix} \cos(\alpha) & \sin(\alpha)\cdot\exp(-i\cdot\phi) \\ \sin(\alpha)\cdot\exp(i\cdot\phi) & -\cos(\alpha) \end{pmatrix}$$

Kui arvuti on piisava mälumahuga ja kiire, et saada sujuvalt hakkama 120 kaadrilise animatsiooniga, vajuta topeltklõps

siia, siin on 0,3 MB fail *Bloch120fr12sec.avi* ja pane animatsioon pidevalt korduma.

Animatsioon kujutab oleku muutumist ajas mingis mikrosüsteemile pidevalt toimivas elektromagnetväljas.

Fikseeritud muutuse saamiseks tuleb rakendada lõpliku kestusega impulsi iseloomuga mõjutust

Näiteks 90-kraadine pööre (180-nene Blochi sfääril) vertikaal ehk meridiaantasandis kirjeldub maatriksiga:

$$U\left(\frac{\pi}{2},0\right) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Tähistame selle operaatori

$$NOT := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Näidata maatriksite peastkorrutamise teel , et NOT-operaator inverteerib omaolekus 1 või 0 oleva kvantbiti omaoleku ja kontrollida väidet töölehearvutuse abil, rakendades alltoodud olekutele operaatorit NOT!

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

NOT-operaator on hästi tuntud klassikalises arvutiteoorias ja -ehistusskeemis ning kui kvantbitid on omaolekus, siis ei erine NOT-operaatori toime (tõeväärtustabel) millegagi klassikalisest.
Kui aga kvantbitt on superpostsiooniolekus, siis muidugi tõeväärtustabel ei koosne enam nullidest ja ühtedest.

Veendu selles , lastes NOT-operaatoril mõjuda superpositsioonolekule

$$|\psi\rangle = \begin{pmatrix} 0.966 \\ 0.148 + 0.212i \end{pmatrix} \qquad \text{NOT}\cdot|\psi\rangle$$

Kuid kvantarvutis saab realiseerida hoopis uusi senitundmatuid loogikalülisid. Näiteks sellist, mis võtaks nagu ruutjuurt eitusest

$$\sqrt{\text{NOT}} := \begin{pmatrix} \frac{1+i}{2} & \frac{1-i}{2} \\ \frac{1-i}{2} & \frac{1+i}{2} \end{pmatrix} .$$

Kontrollime, kas selline on kvantmehaaniliselt realiseeritav, st kas operaator on unitaarne

$$\sqrt{\text{NOT}}\cdot\left(\overline{\sqrt{\text{NOT}}}\right)^{T} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} .$$

Veendume, et operaator õigustab oma nimetust/tähistust, sest kaks lüli järjestikku moodustavad tavalise NOT-lüli:

$$\sqrt{\text{NOT}}\cdot\sqrt{\text{NOT}} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \text{ehk NOT.}$$

Veendu selles, rakendades operaatorit $\sqrt{\text{NOT}}\cdot\sqrt{\text{NOT}}$ olekutele $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$!

Samas, **üheainsa "ruutjuurEIst"** toime isegi omaolekus olevale kvantbitile on midagi täiesti tundmatut klassikalisele loogikale:

$$\sqrt{\text{NOT}}\cdot|0\rangle = \begin{pmatrix} 0.5 + 0.5i \\ 0.5 - 0.5i \end{pmatrix} \qquad \sqrt{\text{NOT}}\cdot|1\rangle = \begin{pmatrix} 0.5 - 0.5i \\ 0.5 + 0.5i \end{pmatrix}$$

Arvuta peast, millise tõenöosusega saadud olek mõõtmisel kukub kokku kvantbiti kummakski omaolekuks ja kontrolli ennast mõõtmiste simulaatori MõõtmistulemusteTNtabel(mingiketvektor) abil.

Üsna ootamatu on aga **"ruutjuurEIst"** toime tulemus "kaldolekule" $|/> = \begin{pmatrix} 0.707 \\ 0.707 \end{pmatrix}$.

Lase töölehel see välja arvutada ja formuleeri tulemuse tähendus!

5. Kahe(kvant)bitised ja universaalsed loogikaelemendid/lülid

Arvutuste teostamiseks vajaliku *network*'i saab kokku panna paarist universaalsest kahe kvantbitiga opereerivast lülist . Üks neist on XOR ehk CNOT (controlled-NOT), mille operaator on

$$\text{CNOT} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Veendu, kasutades transponeerimistehet (Ctrl+1), et CNOT on unitaarne maatriks(operaator)!

Veendu, et CNOT rakendamine on ekvivalentne käsukoodiga
if (|A>=|1>) then |B> ----> NOT|B> , kusjuures |A> jäetakse muutumatuks

$$|00> = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad |01> = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \qquad |10> = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \qquad |11> = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\text{CNOT} \cdot |00> = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Selgita endale CNOT äsjasaadud rakendustulemused, uurides teisenemisvalemit sümboleis:

$$\text{CNOT}\cdot\left[\text{O}\left[\begin{pmatrix}\text{vektAkomp}_1\\ \text{vektAkomp}_2\end{pmatrix},\begin{pmatrix}\text{vektBkomp}_1\\ \text{vektBkomp}_2\end{pmatrix}\right]\right]\rightarrow\begin{pmatrix}\text{vektAkomp}_1\cdot\text{vektBkomp}_1\\ \text{vektAkomp}_1\cdot\text{vektBkomp}_2\\ \text{vektAkomp}_2\cdot\text{vektBkomp}_2\\ \text{vektAkomp}_2\cdot\text{vektBkomp}_1\end{pmatrix}$$

ehk identselt:

$$\text{CNOT}\cdot\begin{pmatrix}\text{vektAkomp}_1\cdot\text{vektBkomp}_1\\ \text{vektAkomp}_1\cdot\text{vektBkomp}_2\\ \text{vektAkomp}_2\cdot\text{vektBkomp}_1\\ \text{vektAkomp}_2\cdot\text{vektBkomp}_2\end{pmatrix}\rightarrow\begin{pmatrix}\text{vektAkomp}_1\cdot\text{vektBkomp}_1\\ \text{vektAkomp}_1\cdot\text{vektBkomp}_2\\ \text{vektAkomp}_2\cdot\text{vektBkomp}_2\\ \text{vektAkomp}_2\cdot\text{vektBkomp}_1\end{pmatrix}$$

6. Kahe kvantbiti ühine faktoriseerumatu ehk põimseisund (entangled state *) ehk põimbitt (* ebit *)*

Veendu, et CNOT rakendamine olekule *AkalduB0*, kus üks kvantbitt on superpos.olekus ja teine omaolekus, annab tulemusena põimseisundi (*entagled state*)

$$\frac{1}{\sqrt{2}}=0.707 \qquad |\!/\!> =\begin{pmatrix}0.707\\ 0.707\end{pmatrix} \qquad \text{AkalduB0} := \text{O}(|\!/\!>,|0>)$$

$$\text{AkalduB0} =\begin{pmatrix}0.707\\ 0\\ 0.707\\ 0\end{pmatrix}$$

$$\frac{1}{\sqrt{2}}\cdot\text{O}(|0>,|0>)+\frac{1}{\sqrt{2}}\cdot\text{O}(|1>,|1>)=\begin{pmatrix}0.707\\ 0\\ 0\\ 0.707\end{pmatrix}$$

Selgita alltoodud sümbolavaldiste paremaid pooli võrreldes, miks **ei ole** CNOT rakendamise tulemusena tekkiv ülaltoodud olek avaldatav kvantbittide olekute otsekorrutisena, st kuidas tekib sõlmitus; vt ka juhtu, kui kvantbitt B on olekus |1> .

$$O\left[\begin{pmatrix}\text{vektAkomp}_1\\ \text{vektAkomp}_2\end{pmatrix},\begin{pmatrix}1\\0\end{pmatrix}\right]\rightarrow\begin{pmatrix}\text{vektAkomp}_1\\0\\ \text{vektAkomp}_2\\0\end{pmatrix}$$

$$\text{CNOT}\cdot\left(\begin{pmatrix}\text{vektAkomp}_1\\0\\ \text{vektAkomp}_2\\0\end{pmatrix}\right)\rightarrow\begin{pmatrix}\text{vektAkomp}_1\\0\\0\\ \text{vektAkomp}_2\end{pmatrix}$$

Kas 2-kvantbitise süsteemi olek $|\psi\rangle := \frac{1}{4}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$ on põimolek?

---------- Töölehe lõpp ----------

(Klassikalisest) krüptograafiast

1. Sissejuhatus

Inimlik soov salasidet pidada on vähemasti niisama vana kui kirjakeel. Seega krüptoloogia juured ulatuvad antiiksetesse tsivilisatsioonidesse Mesopotaamias, Egiptuses, Indias, Hiinas jm.

Umbes nelisada aastat enne Kristust kasutasid Sparta sõdalased krüpteerimisvahendit nimetusega SCYTALE ja tundsid kaht tänapäevani kasutatavat krüpteerimise alusmeetodit – tähtede/sümbolite
substitutsioon ehk asendamine
ja transpositsioon ehk nihutamine.

Arvutis kasutatavad kooditabelid tärkide viimiseks arvulisele kujule on heaks näiteks substitutsioonile. Kooditabel ilmutab end järgmistes *MC* sisseehitatud stringifunktsioonides (vt. *MC Help-Functions*):

$\text{str2vec}("A") = (65)$ $\text{vec2str}((65)) = "A"$

$Teade := "HOMME !"$

$Vektor := \text{str2vec}(Teade)$ $Vektor^T = (72\ \ 79\ \ 77\ \ 77\ \ 69\ \ 32\ \ 33)$

Miks siin kasut. *MC* maatriksoperatsiooni ^{T} ?

$\text{vec2str}(Vektor) = "HOMME !"$

Kollaseks toonitud määratluses **muuda** *Teade*-t ja uuri selle šifreerimist koodivektoriks ja dešifreerimist funktsioonide *string-to-vector* ja *vector-to-string* abil.
Milline arv vastab tühikule (ehk mis on tühiku ASCII-kood)?
Milline on väikseim arv, millele vastab oma sümbol ehk tärk ekraanil?
Kontrolli oma teadmisi kooditabelist järeleproovimistega siintoodud stringifunktsioonide abil.

Rakendame nüüd ka transpositsiooni ehk nihutust – näiteks lahutame igast koodiarvust arvu
$nihe := 22$
Defineerime kodeerimisf.-nid: $Kodeeri(teade) := \text{str2vec}(teade) - nihe$

$Dekodeeri(koodidevektor) := \text{vec2str}(koodidevektor + nihe)$

Olgu tekst näiteks (muuda teksti ja uuri koode):

$Tde := "Kell\ 8.05"$

$KoodVktr := Kodeeri(Tde)$

$KoodVktr^T = (53\ \ 79\ \ 86\ \ 86\ \ 10\ \ 34\ \ 24\ \ 26\ \ 31)$

$Dekodeeri(KoodVktr) = "Kell\ 8.05"$

Paraku ei ole üksikute tähtede arvudega kodeerimine kuigi tõsiseltvõetav shifreerimismeetod. Isegi juhul, kui sala-kommunikeerujad A ja B (teate saatja Alice ja saaja Bob, nagu neid kogu ingliskeelses krüptoloogiakirjanduses on kombeks nimetada) on endile koostanud omast arust väga keerulise kooditabeli, on sellist koodi suhteliselt kerge lahti muukida. Nimelt saab ära kasutada erinevate tähtede tekstides esinemise erinevaid sagedusi igas keeles.

2. Tänapäeval kasutatavatest klassikalistest krüptosüsteemidest

Et paremini mõista kvantkrüptosüsteemide eriomadusi, tutvume enne kahe tänapäeval laialt kasutatava klassikalise šifreerimismeetodiga. Epiteet 'klassikaline' on siin vastandamaks 'kvantmehaanilisele' ega tähenda üldsegi mitte 'vana' vms – vastupidi, näiteks RSA krüptosüsteem leiutati vaid veerandsajandi eest.

2.1 Vernam'i šiffer

Inglise keeles tuntud ka nimetuse all *'one-time-pad' (cryptosystem).*

1. etapp – teate kodeerimine arvudeks toimub samal põhimõttel, mida näitlikustasime eelmises punktis, kust me võtamegi konkreetse kodeerimisreegli ja -funktsioonid:

Alice lähtetekst: Tde := "Raha on kapi taga "

KoodVktr := Kodeeri(Tde)

$KoodVktr^T =$

	1	2	3	4	5	6	7	8	9	10
1	60	75	82	75	10	89	88	10	85	...

Klõpsa tabelil nägemaks kõiki strlen(Tde) = 18 koodi

2. etapp – Alice konstrueerib **juhuslikest** arvudest minimaalse ja maksimaalse koodiarvu vahel **võtme** →), mille pikkus (vektori terminites vektori dimensionaalsus e. komponentide arv) sama, mis kodeeritud teatel.

Pane kursor *Tde*-le ja vajuta F9 saamaks uus juhuslik *Vti* !

$Vti^T =$

	1	2	3	4	5	6	7	8	9	10
1	0	19	57	34	81	17	70	30	9	...

3. etapp – Alice liidab vektorid kokku ja jagab 99-ga, jääkidest moodustubki lõpuks krüpteeritud *Tde*

$$Krpt := \mathrm{mod}(KoodVktr + Vti, 99)$$

$Krpt^T =$

	1	2	3	4	5	6	7	8	9	10
1	60	94	40	10	91	7	59	40	94	...

Veendu sama jrk.n-ga komponentide kõrvutamisega, et *Krpt*-i arvud-komponendid on tõepoolest *Tde* ja *Vti* vastavate komponentide summa jagamisjäägid 99-ga.

Alice saadab *Krpt*-i Bobile (koos võtme järjekorranumbriga, suur hulk võtmeid on juba eelnevalt salaja Bobile antud)

Deshifreerimiseks võtab Bob võtmepakist just sellesama võtme *Vti* ja teeb temaga tagasilahutamise *modulo* 99 :

$$Dekrpt := \mathrm{mod}(Krpt - Vti + 99, 99)$$

ja seejärel Alice käest varem saadud tähtede kodeerimistabeli pöörajaga (vt p.1) **loeb välja salateate**:

Dekodeeri(Dekrpt) = "Raha on kapi taga "

Antud krüptosüsteemis on šifreerimismeetodeina (lisaks asendamisele ja nihutamisele) kasutatud juhuslike arvude viimist koodi, mis on väga efektiivne koodi "segamiseks". Samuti mõjub infot Eve eest peitv filtrina mod-funktsioon, mis on perioodiline ja seega tema pöördf.-n on mitmene funktsioon.

Süsteemi puuduseks on vajadus suurt hulka võtmeid salakanaleid pidi ette edastada.

2.2 Avaliku võtmega RSA krüptosüsteem

1978.a. lõid Rivest, Shamir ja Adleman nimetuse RSA all laialt tuntuks saanud süsteemi, milles võtme pealtkuulamiskindla edastamise/levitamise probleemi pole, sest võti edastatakse avalikult.

Süsteemi põhiidee seisneb järgnevas. Kui Bob tahab **saada** salajasi teateid, loob ta võtmepaari: **avaliku võtme** ja **salajase võtme**. Esimest levitab ta läbi avalike infokanalite, teise aga jätab ainult enda teada. Kui Alice tahab Bobile salateadet saata, šifreerib ta selle Bobi avaliku võtmega ja saadab Bobile, muretsemata kas Eve võib kanalit pealt kuulata. Bob dešifreerib/dekrüpteerib saadud salateate oma salajase võtme abil. Kogu konks on selles, kuidas annab suhestada avalikku ja salajast võtit nii, et süsteem toimiks kiiresti/efektiivselt, kuid oleks turvaline.
Selleks saab kasutada matemaatikas ühesuunalisteks nimetatud funktsioone, mida on lihtne arvutada, kuid mille inverteerimine on väga arvutusmahukas.

Uurime kahe väga suure algarvu korrutamist:

$n := 2593843747457 \cdot 20934834647 \qquad n \to 54301689953167121742679$.

Näeme, et korrutamine käib arvutil praktiliselt hetkeliselt. Aga tehte inverteering, st tegureiks lahutamine võtab väga oluliselt rohkem aega, milles veendu, pannes *n* või tema arvväärtuse siia alla tühja platsihoidjasse ning lugedes rohelise raami kestmise sekundeid hiirega väljaklõpsu hetkest peale.

factor →

Meie näites on *n* 23-kohaline. Arvu pikkuse kasvamisega kasvab tehteks kuluv aeg isegi parima tänapäevase (Lenstra poolt 1993.a.loodud) algoritmiga tegurite leidmiseks lootusetult kiiresti (super-polünomiaalselt). Seega, kui just ei leiutata kardinaalselt uut tüüpi algoritmi, kulub tänapäeva võimsaimail arvuteil üle saja koha pikkade arvude tegureiks lahutamiseks aastaid. Seega on pooleteistsaja kümnendkohaga arvu – ja just selliseid kasutatakse RSA krüptosüsteemi tõsistes praktilistes rakendustes – tundmatuiks tegureiks lahutamine tänapäeval Eve'le kindlalt ülejõu ülesanne.

Tutvume RSA-süsteemiga simulaatoril, kus arusaadavatel põhjustel piirdume suhteliselt väikeste arvudega (mispuhul Eve'le ei valmista mingit raskust teguriteks lahutamise ülesanne kasvõi peast lahendada ja seega mõningase vaevaga ka šifreering lahti muukida).

1. Bob, kes tahab saada salateateid, leiab kaks suurt algarvu *p* ja *q* ning nende korrutise *n* :

$p := 101 \qquad q := 103 \qquad n := p \cdot q \qquad n \to 10403$

(praktikas *p* ja *q* valitakse enam erinevad, et tegureid poleks lihtne leida $n^{1/2}$ lähedusest).

2. Bob valib täisarvu *d*, mis on kaasalgarv korrutisele (*p* -1)(*q* -1) = *r* : $r := (p - 1) \cdot (q - 1)$

$r \to 10200$

Olgu *d* suurem kui 70 $d := \text{Kaasalgarv}(r, 70) \qquad d \to 71$

Funktsiooni Kaasalgarv(,) def.-n ☞

3. Bob leiab väikseima täisarvu *e*, mispuhul korrutis *ed* jagatud *r*-ga annab jäägiks arvu 1 ehk teisiti öeldult – täisarvu *e*, mis oleks *r*-i pöördarv *modulo r* .

$e := \text{mod}\left(d^{-1}, r\right) \qquad e \to \frac{1}{71}$

Kuna viimane MC versioon osutub võimetuks teha modulo-arvutusi, siis on siin sisestatud õige $e := 431$

4. Bob levitab kõigile oma avalikku võtit, mis koosneb arvudepaarist *e* ja *n* ja jätab enda teada salavõtme:

$Avti := \begin{pmatrix} e \\ n \end{pmatrix}$ $Svti := \begin{pmatrix} d \\ n \end{pmatrix}$ $Avti \rightarrow \begin{pmatrix} 431 \\ 10403 \end{pmatrix}$ $Svti \rightarrow \begin{pmatrix} 71 \\ 10403 \end{pmatrix}$

5. Alice kodeerib oma teate arvudeks vahemikus 1...*n*:

Alice lähtetekst: `Tde := "Homme k23 Kassitoomel!"` `KoodVktr := Kodeeri(Tde)`

Klõpsa tabelil nägemaks kõiki strlen(Tde) = 22 koodi

$KoodVktr^T =$

	1	2	3	4	5	6	7	8	9	10
1	50	89	87	87	79	10	85	28	29	...

Selleks sobib näiteks kahekohalised koodid panna paarikaupa kokku, st igale tähepaarile saame vastavusse 4-kohalise arv. Niisugune protseduur on tehtav vahepeal arvud stringideks muundades, misläbi koodide liitmine on teostatav stringide konkaneerimise (aheldamise) tehtega.

$i := 1 .. \frac{strlen(Tde)}{2}$ $KoodVktrStr_i := concat(num2str(KoodVktr_{2 \cdot i-1}), num2str(KoodVktr_{2 \cdot i}))$

$KoodVktrStr^T =$

	1	2	3	4	5	6	7	8	9
1	"5089"	"8787"	"7910"	"8528"	"2910"	"5375"	"9393"	"8394"	...

$KoodVktr4num := \overrightarrow{str2num(KoodVktrStr)}$

$KoodVktr4num =$

	1
1	5089
2	8787
3	7910
4	8528
5	2910
6	5375
7	9393
8	8394
9	8989
10	8779
11	8611

Siinkohal tuleb vasaku vektori komponentarvud kopeerida paremal oleva vektori .....U määratlusse. Kui teate pikkus pole 22, tuleb paremale sisestada (*Insert-Matrix*) uus vektoriplats. Kopeerimiseks klõpsa vasakul 4-kohalisele arvule, siis Ctr+C , siis kursor paremale platsihoidjale ja Ctrl+V.

$KoodVktr4numU := \begin{pmatrix} 5089 \\ 8787 \\ 7910 \\ 8528 \\ 2910 \\ 5375 \\ 9393 \\ 8394 \\ 8989 \\ 8779 \\ 8611 \end{pmatrix}$

6. Alice šifreerib saadud koodi Bobi avaliku võtmega alljärgneva valemi abil ja saadab tulemuse Bobile:

$Evktr := \overrightarrow{mod[(KoodVktr4numU)^e, n]}$

$Evktr_1 \rightarrow 4910$

$Evkr_{,1} \rightarrow Evkr_{,1}$

$Evktr^T \rightarrow (4910 \quad 2424 \quad 7846 \quad 7228 \quad 9817 \quad 8453 \quad 2626 \quad 8088 \quad 606 \quad 8065 \quad 869)$

7. Bob dešifreerib saadud teate oma salajase võtmega alljärgneva valemi abil

$$Svti \rightarrow \begin{pmatrix} 71 \\ 10403 \end{pmatrix}$$

$$DEvktr := \overrightarrow{\text{mod}\left[(Evktr)^{d}, n\right]}$$

Aeglasemal arvutil soovitav kasutada arvutamiseks valemivarianti sissekirjutatud võtmearvudega (klõpsa valemil parema klahviga ja *Enable evaluation*):

$$DEvktr_i := \text{mod}\left[\left(Evktr_i\right)^{71}, 10403\right]$$

$$DEvktr^{T} \rightarrow (5089 \quad 8787 \quad 7910 \quad 8528 \quad 2910 \quad 5375 \quad 9393 \quad 8394 \quad 8989 \quad 8779 \quad 8611)$$

Näeme, et Alice teate kood (vt siin all korratuna) on taastatud!

$KoodVktrStr^{T} =$

	1	2	3	4	5	6	7	8	9
1	"5089"	"8787"	"7910"	"8528"	"2910"	"5375"	"9393"	"8394"	...

Kui peame silmas, et tegelikes RSA rakendustes on need arvud väga palju pikemad, siis selline dekodeerimine Eve poolt, kes ei tea salavõtit, on võimalik vaid siis, kui ta suudab leida arvu *n* tegurid ja seeläbi arvutada välja õige täisarv *d* dešifreerimisvalemile.

7. Bob dekodeerib koodid teateks kodeerimistabeli abil.

Kuna RSA-süsteemi olulisim punkt oli eelmine ja edasine numberkoodist tähtsõnumi taastamine ei erine põhimõtteliselt eelmises paragrahvis tehtust, siis siinkohal loeme šifreerimis-dešifreerimis-ülesande lahendatuks.

Juba enne akadeemilise autorikollektiivi RSA kuulsat publikatsiooni 1978.a. oli Briti valitsuse vastavates agentuurides tuntud – ehkki see sai ametlikult teatavaks alles hiljuti – nn *Non-Secret Encryption*, mille leiutas 1970.a. J.Ellis ja arendas edasi nüüd RSA nime all tuntud süsteemiks C.Cocks.

Moraal: kes on auahne ja soovib oma avastusega tuntuks saada, peab hoiduma tööleasumisest kinnistes/salajastes uurimiskontorites.

Probleemülesanne (*for advanced users of MC*).
Lisa töölehele p.7 täitmiseks vajalik.
Näpunäide: võimaluse 4-kohaliste koodide 'tagasihakkimiseks' leiab *MC* stringifunktsioonide hulgast.

3. Lõpetuseks

Vernami vms šifri puhul probleemiks jääv võtme turvaline edastamine langeb RSA avaliku võtmega süsteemis hoopis ära. Kuid – meenutame loengukursuse sissejuhatavat presentatsiooni – väga populaarseks saanud RSA turvalisust ohustab 1996.a. P.Shor'i poolt loodud kvantarvutuslik algoritm suurte arvude tegurite leidmiseks. Shori algoritmile on pühendatud eraldi mahukas tööleht.

Kui kvantfüüsika ühe käega ähvardab põhja lasta RSA kui tänapäevase krüptograafia lipulaeva, siis teise käega pakub ta – või täpsemini, on juba pakkunud – omaltpoolt asemele uue: **kvantkrüptograafia**. Sajandivahetuseks ekperimentaalselt realiseeritud kvantkrüptograafia kahele meetodile on pühendatud eraldi tööleht.

---------- Töölehe lõpp ----------

Kahe kvantbitt-registriga arvuti simulatsioon Shor'i algoritmi abil tegurite leidmiseks perioodilise funktsiooni perioodi määramise kaudu

Viimati täiendatud
30.08.2011

Alustuseks käsitleme veelkord põimimata/põimitud (*entangled*) olekute eristamist ja Einsteini sõnul "õudsa (*spooky*) kaugmõjuni" viivat mõõtmisakti liitsüsteemis.

1. Kahest kvantbitt-registrist koosneva liitsüsteemi oleku osaline kollapseerumine ehk väljaprojekteerumine mõõtmisaktis, mida tehakse ainult ühe registri kvantbittidel.

Olgu meil kaks registrit A ja B , üks koosnegu n_A kvantbitist, teine n_B kvantbitist. Olgu konkreetselt edaspidise graafilise näite jaoks need arvud hästi väikesed: $n_A := 6$ ja $n_B := 4$.

Tähistame kumbagi registri omaolekuid neisse kirjutatud kümnendarvuga a ja b , mis võivad omandada väärtusi

$a := 0 .. \left(2^{n_A} - 1\right)$ $b := 0 .. \left(2^{n_B} - 1\right)$ kus $N_A := 2^{n_A}$ $N_A = 64$ $N_B := 2^{n_B}$ $N_B = 16$

Sellel töölehel me loendame vektorite ja maatriksite veergusid/ridu alates 0-st, mitte 1-st, et loendamine langeks kokku kümnendarvudega, mida registri omaolekud kannavad.
(vt menüüst *Math:Options:Array Origin*, = 0 , mida mitte muuta !)

Kumbagi registri suvaline (superpositsioon)olek avaldub (üldiselt kompleksarvuliste) vektorikomponentide ja oma-olekuvektorite kaudu (vt. sissejuhatuse tööleht):

$$|A\rangle := \sum_a \left(A_a \cdot |a\rangle\right) \qquad |B\rangle := \sum_b \left(B_b \cdot |b\rangle\right)$$

ja mõlema registri kui liitsüsteemi olek avaldub otsekorrutisena (NB! siin "esimese" korrutatava paneme paremale):

$$|\psi\rangle = |B\rangle \otimes |A\rangle = \left[\sum_b \left(B_b \cdot |b\rangle\right)\right] \otimes \left[\sum_a \left(A_a \cdot |a\rangle\right)\right] = \sum_b \sum_a \left[\left(B_b \cdot A_a\right)\left(|b\rangle \otimes |a\rangle\right)\right] \quad (1)$$

Kirjutame selle võrduste rea kolmanda kuju registri B osas lahti ilma summamärgita nii, et toome esile jooksva naturaalarvu/indeksi b mingile ühele väärtusele b=b'=8 (näiteks) vastava liidetava superpositsioonisummas ja ülejäänuid tähistame punktireaga:

$$|\psi\rangle = \left(.... + B_8 \cdot |8\rangle + ...\right) \otimes \left[\sum_a \left(A_a \cdot |a\rangle\right)\right]$$

Sellest kirjutusviisist on otse näha, et kui üks registritest -- B näiteks -- allutada mõõtmisaktile, kus tema supepositsiooniolek kukub kokku üheks omaolekuks - näiteks sisule "8" vastavaks -- siis registri A olekuga ei juhtu midagi ja liitolek pärast mõõtmist avaldub A-registri säilinud oleku ja B-registri omaoleku |8> otsekorrutisena (liitoleku õigeks normeeringuks tuleb komponent B_8 asendada 1-ga). Paneme tähele, et A allesjääv seisund **ei sõltu B-registri mõõtmise konkreetsest tulemusest** (st **mis** olekusse -- kas |8> või mõni muu -- register B läks).

See on loomulik tulemus, sest me vaatlesime mõõtmist vaid ühe liitsüsteemi olekute **eriliigi** puhul. Üldjuhul pole aga liitsüsteemi olek taandatav (st faktoriseeruv) osasüsteemide olekute otsekorrutiseks. Nagu sissejuhataval töölehel juba uurisime, on üldjuhul liitsüsteemi olek osasüsteemide olekute suhtes põimolek (*entangled state*). Põimolekus liitsüsteemi ühe osasüsteemi mõõtmisaktis ei jää teise osasüsteemi olek puutumata, mis viibki kuri-kuulsale korreleeritusenähtusele -- Einsteini ristitud "õudsele (*spooky*) kaugmõjule".

Antud juhul ja kasutusele võetud tähistustes on kahe registri kui liitsüsteemi olek üldjuhul, nagu ikka kvantmehaanikas, superpositsioon-lineaarkombinatsioon kõigist $N_A \cdot N_B = 1.024 \times 10^3$ liitsüsteemi omaolekuvektorist, kusjuures iga omavektori ees on üks võimalikust $N_A \cdot N_B = 1.024 \times 10^3$ erinevast ja omavahel sõltumatust (v.a. tavaline normeeringutingimus, et moodulite ruutude summa peab olema 1) kompleksest koefitsiendist C. Allpool me näeme, et on ülevaatlikum, kui mitte nummerdada superpositsiooni vektorikomponente C (ega ka liitsüsteemi omaolekuid ja omavektorite endi komponente) ühe indeksiga vahemikus 1.... $N_A \cdot N_B = 1.024 \times 10^3$, vaid säilitada (kokkuvõttes sama arvu andev) **kaheindeksiline tähistusviis** indeks-täisarvude a ja b abil.
Niimoodi -- kirjutades ka liitsüsteemi omavektorid registrite A ja B omavektorite korrutise kujul, saame avaldada liitsüsteemi üldoleku kujul:

$$|\psi\rangle = \sum_b \sum_a \left[C_{b,a} \cdot (|b\rangle \otimes |a\rangle) \right]$$

Vektorikomponentide üleskirjutamine kaheindeksilise maatriksi kujul (üheveerulise vektori asemel) toob ilmekalt välja meile juba esimesest töölehest tuttava erinevuse põimoleku ja põimimata ehk otsekorrutis-oleku vahel. Nimelt , võttes ülalt valemist (1) erikuju koefitsientidele C, st $C_{b,a} := B_b \cdot A_a$, näeme, et valemis (1) on koefitsiendi maatriks C faktoriseeruv ehk n.-ö "vaese sisuga" , st C on ridu- ja veergepidi vastavalt proportsionaalne ja kõik tema elemendid on moodustatud vaid $N_A + N_B = 80$ erinevast arvust $N_A \cdot N_B = 1.024 \times 10^3$ asemel.

Olgu meil näitena register A olekus, kus erinevad arvud on jaotunud ühekülgse (loogilise funktsiooni $(a \geq 25)$ abil) eksponentsiaalse jaotuse kohaselt, kusjuures maksimaalse amplituudiga olgu registris arv 25, aga registris B olgu arvude amplituudid võrdelised arvu endaga (1-st lugedes), siis registrite olekute normeeritud vektorikomponendid avalduvad:

$$\text{normi}_B := \sum_b \left(\left| b + 1 \right| \right)^2 \qquad B_b := \frac{b + 1}{\sqrt{\text{normi}_B}}$$

Kontrollime tõenäosuste normeeritust: $\sum_b \left(\left| B_b \right| \right)^2 = 1$

Milleks on meil vaja ette arvutada abisuurus "normi" ?

$$\text{normi}_A := \sum_a \left[\left| (a \geq 25) \cdot \exp\left[\frac{-(a - 25)}{5} \right] \right| \right]^2 \qquad A_a := \frac{(a \geq 25) \cdot \exp\left[\frac{-(a - 25)}{5} \right]}{\sqrt{\text{normi}_A}}$$

$$\sum_a \left(\left| A_a \right| \right)^2 = 1$$

Defineerime nüüd C:

$C_{b,a} := B_b \cdot A_a$ Muuseas, selliselt saadud maatriksi elementide sümmeetria/sõltuvus avaldub selles, et maatriksi astak on madalaim võimalik, st 1 : $\text{rank}(C) = 1$

Rida-veeruti proportsionaalsust on hea tuvastada ka graafiliselt.
Klõpsates all pildil ja hiirega tirides teda erinevate vaatenurkade alla, on kerge veenduda, kuidas read ja veerud on vastavalt proportsionaalsed, nagu peabki olema faktoriseeruval kahe-muutuja funktsioonil (antud juhul muutujad diskreetsed , st indeks-muutujad).

C

Kui 3D graafiku pilt jääb osaliselt või tervenisti mustaks ja/või kaob hoopis ära töölehe tagasikerimisel ning pildi kerimine ekraani keskele ja/või hiireklõps graafikul asja ei paranda, on tõenäoselt tegemist antud arvuti graafikakaardi draiveri vms (installatsiooni/versiooni) probleemidega.
Kui Win 7 all 3D pilt kaldub valgeks minema, klõpsa temal ja/või loe sellekohast juhist kursuse pealehel või **siin** (tee topeltklõps lingil).

Graafikut võib tema paremaks vaatlemiseks pöörata, teisiti värvida või valgustada jne, kui teha pildil topeltklõps ja muuta arve jm avanenud menüüs.
Pildi võib panna koguni **pöörlema** -- näiteks ümber vertikaali -- kui viia kursor graafiku keskele, vajutada alla *Shift*-klahv ja , hoides vasakut hiireklahvi all, tirida kursorit paremale ning seejärel klahvid vabastada.
Mida kaugemale tirida kursorit enne klahvide vabastamist, seda suurema sammuga tuleb pöörlemine. Kui *Shift*-klahvi mitte all hoida, jääb graafik tirituks uue nurga alla ilma et ta pöörlema hakkaks.
Pöörlemise seiskab hiireklõps pildil.

Graafikut ei ole soovitav jätta "klõpsatud" seisu (milles nähtav "karvane" raamjoon), sest see raskendab valemite ja algandmete muutmist töölehel. Piisab, kui korraks klõpsata vasaku hiireklahviga, kursoriristi olles mingil puhtal vabal pinnal töölehel.

Graafiliselt vastab ülalsaadud tulemusele -- et otsekorrutisolekus liitsüsteemi ühe osa peal mõõtmine jätab teise osasüsteemi oleku muutumatuks -- järgmine operatsioon: B-registrist mingi arvu väljalugemisele vastab graafikul ainult ühe riba 16-st allesjätmine. Iga selline riba aga kordab A-registri olekut normeerimiskonstandi täpsusega. Sama kehtib B-registri kohta mõõtmise teostamisel A-registris.

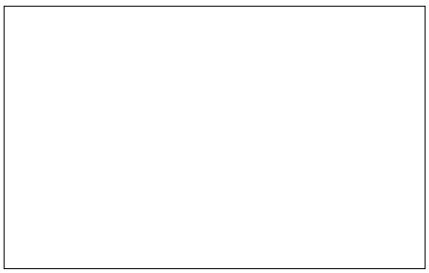
Üldisele olekule -- põimolekule -- vastava maatriksi C kujutises niisugust sümmeetriat pole -- igal B mõõtmisel maatriksist C n.-ö. väljaprojekteerunud veerul/ribal võib olla erinev koefitsientide jaotus piki A-registri "telge" . Konstrueerime nüüd mingi põimoleku maatriksi C_{entgld}, kaotades veergude nihutamisega ridadevahelise proportsionaalsuse C-s. Allpool me keerame maatriksi "pealtvaatesse", st koefitsientide väärtusi markeerime ainult värvitooniga.

$$\mathrm{rank}(C) = 1$$

$$C_{entgld_{b,a}} := C_{b,\mathrm{if}(a \geq 15, a-b, a)}$$

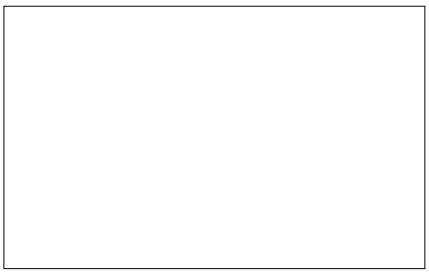
$$\mathrm{rank}\left(C_{entgld}\right) = 16$$

Näeme, et astak omandas nüüd 1-st suurema väärtuse (antud juhul koguni maksimaalselt võimaliku väärtuse 16-realise maatriksi jaoks)



C

Vasakpoolsel pildil on jällegi tuvastatav, et amplituudide jaotus piki erinevaid ridu on identne, st erineb vaid värviskaala intensiivsuse (st. nagunii uuesti normeeritava kordaja) poolest. Ükskõik, mis on B mõõtmise tulemus, st ükskõik missugune rida jääb alles peale liitregistri seisundi "väljaprojekteerumist" B osas, allesjääv A amplituudide jaotus, seega allesjääv A olek, on ikka üks ja seesama. Sama kehtib veergude suhtes, st järeldus on sama, kui mõõta A-d ja allesjääv on superpositsioon B-s).



C_{entgld}

Kui nüüd siin B-d mõõta, siis erinevate mõõtmistulemuste puhul saame ka erineva amplituudide jaotusega superpositsioonseisundi registris A. Näiteks, kui B mõõtmisel saadi "8", siis piki allesjäävat 8-ndat rida on amplituudide jaotus erinev jaotusest piki suvalist teist rida, näiteks 5-ndat, mis vastaks B-registri mõõtmistulemusele "5". Antud juhul see erinevus seisneb küll ainult maksimumi nihkes ja jaotuse kuju ise on normeerimisteguri täpsusega sama kõikides ridades, kuid sellest piisab kaotamaks reati-veeruti proportsionaalsust, mis oli omane mittepõimitud seisundile.

Kontrollime, kas normeering on O.K.:

$$\sum_b \sum_a \left(\left| C_{entgld_{b,a}} \right| \right)^2 = 1$$

Nii nagu valemi (1) järgi uurisime otsekorrutis-seisundi väljaprojekteerumist B-registri mõõtmisel, vaatame siin üldise põimseisundi väljaprojekteerimist. Oletame näiteks jällegi, et mõõtmise peale kukkus B- register kokku omaolekusse "8". Siis saame kirjeldada seda väljaprojekteerumist valemite keeles nii:

$$|\psi> = \sum_b \sum_a \left[C_{b,a} \cdot (|b> \otimes |a>) \right]^{\blacksquare} \quad \text{--------->} \quad |\psi> = |8> \otimes \left[\sum_a \left(C_{8,a} \cdot |a> \right) \right]^{\blacksquare}$$

Saadud avaldises tuleb vaid kõik amplituudid läbi korrutada konstandiga, et taastada olekuvektori normeering.

Amplituudid $C_{8,a}^{\blacksquare}$ mängivad A-registri jaoks B mõõtmise tagajärjel A-registris tekkinud superpositsiooni amplituudide rolli (vt $A_a^{\blacksquare}$ paragrahvi alguses). Jällegi näeme, et niipea, kui maatriksi C kaheksas rida erineb suvalisest teisest reast enam kui lihtsalt proportsionaalsus-kordaja poolest, hakkab A-registri seisund sõltuma mõõtmistulemusest B-registris. Mõõtmise ajaks võivad mõlemad registrid olla üksteisest väga kaugele liikunud, ikkagi B mõõtmine mõjutab hetkeliselt A-registri seisundit. See ongi see "õudne kaugmõju", mis aga siiski ei tähenda informatsiooni hetkelise (või ülevalgusekiirusega) ülekande võimalikkust.

Toome veel näite kahe ühekvantbitise registriga ja vaatame, kuidas **maatriksi** C omadused ilmutavad põimitust või mittepõimitust ka selle lihtsaima süsteemi näitel ehk ilmekamaltki, kui liitsüsteemi **olekuvektori veeru** uurimine, mida tegime **sissejuhatava töölehe (1_RegisterjaGateMC8.mcd) lõpus**

Paneme: $n_A := 1$ $n_B := 1$ ja kordame lehe algusest $a := 0 .. \left(2^{n_A} - 1\right)$ $b := 0 .. \left(2^{n_B} - 1\right)$

Olgu $A := \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ ja $B := \begin{pmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{pmatrix}$ Siis otsekorrutisseisundi maatriks $C_{b,a} := B_b \cdot A_a$ võrdub $C := B \cdot A^T$

$C =$

	0	1
0	0.866	0
1	0.5	0

C pildina:

Näeme, et teise rea saame esimesest, kui seda korrutada konstandiga $\frac{0.5}{0.866} = 0.577$ ja astak $\text{rank}(C) = 1$

Vektorina kirjutatult on $|B\rangle \otimes |A\rangle =$ $\begin{pmatrix} 0.866 \\ 0 \\ 0.5 \\ 0 \end{pmatrix}$ ehk kasutades 2-bitiste registrite omaolekute (vt. äärise taga -->) kahendarv-tähistusi $|00\rangle := |0\rangle \otimes |0\rangle$ jne, kus "vanem" bit B vasakul, "noorem" A-bit paremal, see seisund avaldub ka kujul

$|\psi\rangle := \begin{pmatrix} 0.866 \\ 0 \\ 0.5 \\ 0 \end{pmatrix}$ $|\psi\rangle =$ $0.866 \cdot |00\rangle + 0.5 \cdot |10\rangle =$

	0
0	0.866
1	0
2	0.5
3	0

Kui aga nüüd tekitada ostekorrutis-seisundist näiteks lüli CNOT abil põimseisund (vt ja võrdle **sissejuhatava töölehe (1_RegisterjaGateMC8.mcd)** §5, §6) mille veeru tükeldamisel A - ja B ridadeks-veergudeks saame maatriksi

$C_{entgld} := \begin{pmatrix} 0.866 & 0 \\ 0 & 0.5 \end{pmatrix}$, mille read enam pole proportsionaalsed ja astak ei ole enam 1 $\text{rank}\left(C_{entgld}\right) = 2$

$|00\rangle := \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ $|01\rangle := \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$

$|10\rangle := \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$ $|11\rangle := \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$

$CNOT := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

$CNOT \cdot |\psi\rangle =$

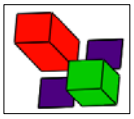
	0
0	0.866
1	0
2	0
3	0.5

,

$0.866 \cdot |00\rangle + 0.5 \cdot |11\rangle =$

	0
0	0.866
1	0
2	0
3	0.5

C_{entgld} pildina:



C_{entgld}

Kuna $CNOT \cdot |\psi\rangle$ on superpositsioon kujul $CNOT \cdot |\psi\rangle \equiv 0.866 \cdot |00\rangle + 0.5 \cdot |11\rangle$, siis B mõõtmisel lugem "0" (tns.=3/4) määrab ära ka A-registri oleku -- antud juhul viib ka A omaolekusse "0". B lugem "1" (tns.=1/4) aga viib A samuti omaolekusse "1" . Sarnaselt põimitud footoneid ja nendevahelist korrelatsiooni mõõtmiste suhtes ehk sedasama kurikuulsat " õudsat kaugmõju" kasutatakse kvant-krüptograafias, kvant-teleportatsioonis jm.

Arusaamatuste vältimiseks märgime lõpuks, et

☞ erinevalt **sissejuhatava töölehe (1_RegisterjaGateMC8.mcd)** kasutusest ning m.h.§ 5-s toodud CNOT selgitusest tingimuslause 'if --->then' abil, on siin töölehel A ja B **järjestus vastupidine;**

☞ maatriks C pole muud kui liitsüsteemi vektori komponentide ristkülikukujulise paigutuse tulemus, sellel maatriksil ei ole mingit iseseisvat uut sisu ega tähendust, ning teda **ei tohi segamini ajada tihedusmaatriksiga,** mis konstrueeritakse samuti olekuvektori komponentidest.

Kuna *bra*-vektor on kaaskompleksi reaks keeratud *ket*, st. $\langle\psi := \overline{\left(|\psi\rangle^T\right)}$ ehk

$\langle\psi =$

	0	1
0	0.866	0

siis skalaarkorrutis $\langle\psi \cdot |\psi\rangle = 1$ näitab lihtsalt normi, aga

nn väliskorrutis $|\psi\rangle \cdot \langle\psi =$

	0	1	2	3
0	0.75	0	0.433	0
1	0	0	0	0
2	0.433	0	0.25	0
3	0	0	0	0

on a
tihe

Ja nüüd siis asume selle (kuri)kuulsa, arvutiteaduses torme tekitanud algoritmiga tutvuma...
tavaarvutil simulatsiooni ja graafikute abil.
Võta kõrvale algoritmi etappide ingliskeelne loetelu.

2. Shori algoritmi simulatsioon

Seda töölehte on täiendanud võimalusega simuleerida perioodi leidmist mitmel erineval *x* ja *n* väärtusel kursuse 2005.a. kuulaja Heli Valtna.

Kaheks teguriks lahutatav arv on *n*, mis simulaatoris on krüptimises tegelikult kasutatavaist võrreldamatult väiksem:

$n := 15$ ja mille tegurid juba n.ö. palja silmaga näha: 15 = 3x5, 21 = 3x7 jms.

Võtame mingi arvu $q := 243$, mis oleks piisavalt palju suurem n-st: $n^2 < q < 2n^2$ (või, teistes allikates, kuni $3n^2$)

Võtame juhuslikult valitud arvu *x*, mis oleks kaasalgarv *n* suhtes

$x := 13$.

Kaasalgarvud on täisarvud, millel suurim ühisjagaja (ühistegur) on 1. Näiteks arve 9 ja 35 nimetatakse kaasalgarvudeks.

Funktsiooni f_{xamodn} uurimiseks laseme tema argumenti joosta lõigul, mille $pikkus := 20$ $z := 0 .. pikkus$

Moodustame jagamisjääki arvutava funktsiooni: $f_{xamodn}(z) := \mathrm{mod}(x^z, n)$

Vasakul, st põhilehel hakkab meil argumenti tähistama *a* *z*-i asemel.

Olgu meil 2 registrit A ja B, milledesse võib kirjutada kümnendarve vahemikes, vastavalt, $a := 0 .. q - 1$ $b := 0 .. n$

Nagu jagamisjäägina määratud funktsiooni jaoks peabki olema (miks?), näeme alltoodud püsttabelist, et väärtused hakkavad perioodiliselt korduma.

Olgu A-registrisse pandud üks funktsiooni argumendi konkreetne väärtus, näiteks kümnendarv 8.
Siis A-registri olekuvektori komponendid avalduvad:

$RegA_a := \delta(a, 8)$

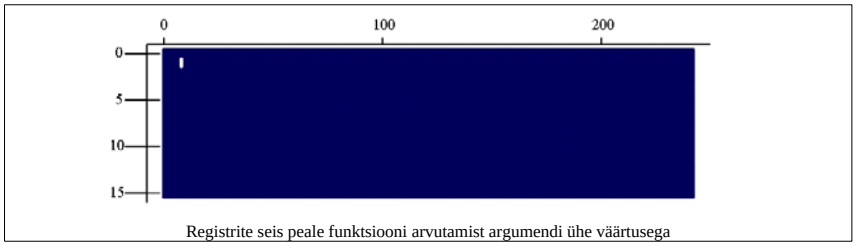
B-registris needsamad aga f-i arvutusvalemile vastavalt:

$RegB_b := \delta(b, f_{xamodn}(8))$

Delta kohta vt.*Help*'ist KRONECKER DELTA

Mõlema registri kui liitsüsteemi ühine olek aga: $Reg_{b,a} := RegB_b \cdot RegA_a$ siin ülevaatlikkuse huvides kujutame liitregistri sisu/olekut maatriksina (nagu ülal §1-s kirjeldati)

Kuna lähteolek polnud superpositsiooniolek, siis ka tulemusena saame liitregistri omaoleku, st ainult üks vektori komponent on nullist erinev : $Reg_{f_{xamodn}(8), 8} = 1$



Registrite seis peale funktsiooni arvutamist argumendi ühe väärtusega

Reg

Nüüd aga asume kvantkompuutri erinevust klassikalisest ära kasutama ja paneme kõigepealt A-registrisse ühtlase superpositsiooni kõigist kümnendarvudest, st kvant-parallelismi abil kõikvõimalikud argumendiväärtused: $C_{Asile} := \frac{1}{\sqrt{q}}$

$f_{xamodn}(z) =$

1
13
4
7
1
13
4
7
1
13
4
7
1
13
4
...

Kuna *n* on siin kõigest 2-kohaline arv, leiab tavaarvuti lihtsal viisil perioodi *r*, näiteks niimoodi:

```
r := loendaja ← 1
     for i ∈ 0 .. pikkus
         samm ← loendaja  if f_xamodn(loendaja) = f_xamodn(0)
         loendaja ← loendaja + 1  otherwise
     error("periood pole paarisarv")  if mod(samm, 2) ≠ 0
     return samm
```

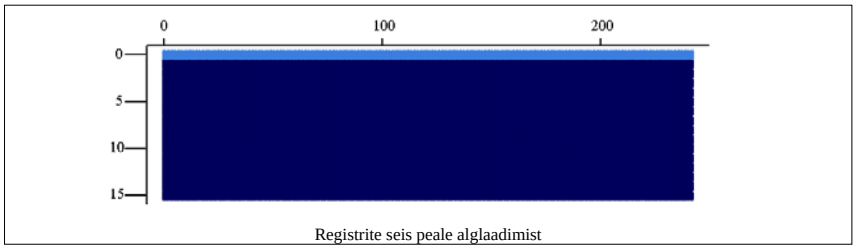
Valitud *n* ja *x* väärtuste puhul on periood $r = 4$ **Kui see on paaritu arv, tuleb valida muu *x* .**

NB! Kui n kohtade arv on saja suurusjärku, nagu RSA protokollis, siis perioodi leidmine ongi faktoriseerimis-algoritmi raskeim etapp, just see, mis vajab kvantarvutit. Meie simulaatoris peame me perioodi ette teadma.

Mis real/veerul joonisel asub valge ristkülik? Miks valge?

$RegA_a := C_{Asile}$ $RegB_b := \delta(b, 0)$ $Reg_{b,a} := RegB_b \cdot RegA_a$

Kui suur on A-registri seisundite Hilberti ruumi dimensioon?



Registrite seis peale alglaadimist

Miks on rida b=0 sinine? Miks kõik teised read on mustad?

Reg

Toimugu nüüd kvantkompuutris olekuvektori samasugune pööre funktsiooni *f* arvutamiseks, nagu varemgi. See pöördeoperatsioon kvantkompuutris annab aga nüüd tulemuse korraga kogu argumentide massiivi jaoks.

Kuna meil on kvantkompuutri **simulatsioon,** siis selle tegemiseks peame konstrueerima tulemuse *Mathcad*'ga, muuhulgas kasutame ära teabe tegelikus probleemis alles otsitava tundmatu suuruse -- perioodi --- kohta:

ning arvutame selle teadmisega välja **maatriksipildi**, mis vastab kvantkompuutri registrite põimolekule peale kvantkompuutris funktsiooni *f* ühekordse arvutusoperatsiooni läbitegemist.

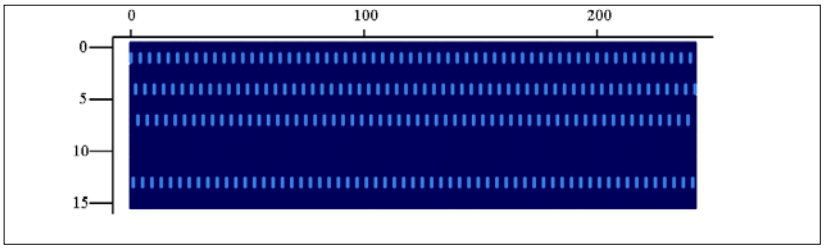
$$Reg_{b,a} := C_{Asile} \cdot \sum_{i=0}^{r-1} \left[\delta\left(b, f_{xamodn}(i)\right) \cdot (\mathrm{mod}(a, r) = i) \right]$$

$$RegB_b := \sum_{i=1}^{r} \left(\delta\left(b, f_{xamodn}(i)\right) \right)$$

Vrdl RegA/B avaldistega eespool.

Mis valemiga me simulaatoris registrimaatriksi täidame, on pigem MC-köögist vajaliku leidmise küsimus, mitte sisuliselt tähtis. Aga kasulik on ikkagi neist valemeist aru saada...

Rõhutame, et see arvutusvalem siin ei kirjelda kvantarvutis toimuvat unitaarset pööret, vaid üksnes konstrueerib pöörde tulemuse. Liigne kord rõhutame ka, et registri(te) superpositsioonolekut "vaadata" tegelikkuses ei saa.



Reg

Miks siin pole helesinine triip pidev nagu eelmisel pildil? Miks on katkendlikud ribad üksteise suhtes horisonaalnihkes ja mida see nihe näitab?
Selgema pildi saamiseks vorminda pilti y-telje skaala maksimumiks väiksem arv, näit. 20. Pane kõrvalolevale fxamodn(z) -le võrdusmärk -- nagu juba tegime ülalpool -- ja võrdle arve pildiga.

$f_{xamodn}(z)$

Kõigi seniste 3D graafikute parameetreid (asendit, värviskaalat, jne) on võimalik parema ettekujutuse saamiseks oma maitse järgi (kuid ühtemoodi!) muuta. Mustvalge väljaprindi jaoks tuleks valida halltoonide skaala z-teljele, anda seal ette väärtuste vahemik -1 kuni 0 ning maatriksi Reg asemel lasta kujutada -Reg .

Paneme tähele, et liitregistri olek on põimolek -- kõiki pildiridu pole võimalik üksteisest saada konstandiga 0 või 1 või vms läbikorrutades, sest heledate ristkülikute read (amplituudide jaotused A-s) **on nihkes** eri ridades.

Teeme B-registri oleku mõõtmise akti simulaatori , kus olek kollapseerub võrdtõenäoselt üheks *r*-st võimalikust.

$$\text{RegB\_lugem} := \left| \begin{array}{l} \text{jumalavalik} \leftarrow \text{rnd}(1) \\ \text{for } i \in 0 \,..\, (r-1) \\ \quad \text{return } f_{\text{xamodn}}(i) \text{ if } \frac{1}{r}\cdot i < \text{jumalavalik} \le \frac{1}{r}\cdot(i+1) \end{array} \right.$$

Miks ainult *r* võimalust, B-l on ju 16 omaolekut?

rnd(1) on *MC* funktsioon, mis annab pseudojuhuslikke arve vahemikus 0.. 1
Pane kursor rnd(1)-le ja klõpsa klahvi F9!

rnd(1) = 0.193

Mõõtmisaktis juhusliku tulemuse saamiseks pane kursor funktsiooni 'RegB_lugem' definitsioonile ja vajuta klahvi F9!

RegB_lugem = 1

Liitregister peale B osas lugemist on väljaprojekteerunud (alles jääb 1 rida, mille kirjutame 1-veeruliseks vektoriks, mis ta ongi):

$$\text{Reg} := \left(\text{Reg}^{T}\right)^{\langle \text{RegB\_lugem} \rangle}$$

Arvutame normeerimiskonstandi vektori sklaarkorrutise iseendaga kaudu:

$$A := \sum \left[\overrightarrow{\left(\left|\text{Reg}\right|\right)^{2}}\right] \qquad A = 0.251 \qquad C := \frac{1}{\sqrt{A}}$$

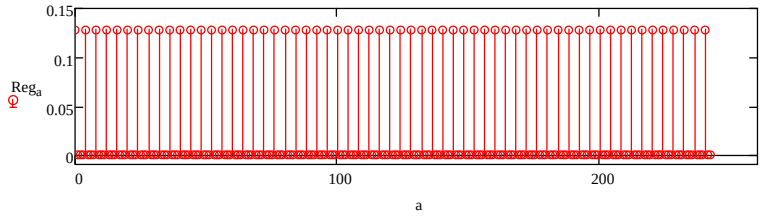
Renormeerime olekuvektori st taastame tema normeerituse 1-le:

$$\text{Reg} := \text{Reg}\cdot C$$

Pane siia alla suurusele Reg juurde märk "=" ja ennusta, mis tuleb!

Reg

Alloleval registrisisugraafikul perioodilisuse peenemaks uurimiseks klõpsa 1x pildil ja vali menüüst *Format/Graph/Zoom* ning moodusta hiirega vedades graafiku keskel 3-5 püstkriipsu ja alusjoont haarav kast, seejärel klõpsa nuppu *Zoom*
Vajadusel otsi abi *Help*'i indeksist võtmega *zoom*.



Mõõtmisprotsessi modelleerimiseks on eelkõige vaja tõenäosusjaotusi, mis mäletatavasti avalduvad amplituudide moodulite ruutudena:

$$\text{MõõtmistulemusteTNtabel(ket)} := \left[\overrightarrow{\left(\left|\text{ket}^{T}\right|\right)^{2}}\right]$$

Sellise registri lugemisest(=mõõtmisest) ei ole kasu, sest me peaksime tegema tohutu arvu korduvaid katseid (iga kord algusest peale), et mõõtmistulemuste statistikast aru saada, et just iga *r*-s arv saab olla registri lugemiks tema mõõtmisel, st et arvutatud funktsiooni periood on *r*.
Perioodi väljatoomiseks rakendame A-registri olekule FFT-teisenduse (unitaarset) operaatorit:

$FourierReg := cfft(Reg)$

vt. *Help*'ist *Fourier Transforms*
Kvantregistri sisule Fourier' teisenduse tegemine on igati "kvantarvutilik" operatsioon.

Registri Fourier-teisendatud olekuvektor annab aga harv-üksikute ja (NB!) perioodile vastavate maksimumidega tõenäosustejaotuse:

$RegAlugemiTnsused := MõõtmistulemusteTNtabel(FourierReg)$

Uuri seda tabelit siin all (teda kerides) ja ennusta, millal tuleb järjekordne maksimum!

--->

$RegAlugemiTnsused =$

	0	1	2	3	4	5
0	0.251	$4.22 \cdot 10^{-6}$	$4.231 \cdot 10^{-6}$	$4.248 \cdot 10^{-6}$	$4.273 \cdot 10^{-6}$	...

Pea alati meeles: "mõõtmisel" kollapseerub olek n.-ö. jumala juhuslikult üheks sellele mõõtmisele omasest omaolekust ning see juhtub tõenäosusega, mis võrdub mooduli ruuduga selle omaoleku amplituudist mõõtmiseelse oleku arenduses (superpositsioonis) omaolekute järgi.

------>
Meil on vaja ka mõõtmisakti simulaatorit, mis juhuslikult valiks ühe punkti alloleval tulemuste tõenäosuste jaotusgraafikul

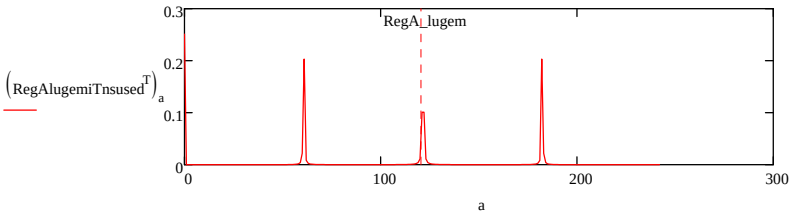
$$OmaolekuNr\_kuhukukubOlekmillel(TNtabel) := \begin{| jumalatäringuvise \leftarrow rnd(1) \\ s \leftarrow TNtabel_0 \\ j \leftarrow 0 \\ \text{while } s < jumalatäringuvise \\ \quad \begin{| j \leftarrow j + 1 \\ \quad s \leftarrow s + TNtabel_j \\ j \end{|$$

rnd(x) on MC pseudojuhuslike arvude tekitajafunktsioon vahemikus 0..x

$RegA\_lugem := OmaolekuNr\_kuhukukubOlekmillel\left(RegAlugemiTnsused^T\right)$

Praegu tuli selle **juhuslikuks** väärtuseks: $RegA\_lugem = 121$

Hoia kursor fuktsiooni 'RegA_lugem' definitsioonil ja, vajutades F9, uuri graafiku muutuse järgi f.-ni tööd



Topeltklõps graafikul ja pane y-skaala logaritmiseliseks!

Hoides kursorit 'RegA_lugem'-i definitsioonil, on vajutustega F9-le võimalik veenduda, kuidas lugem kukub juhuslikult täisarvule, mis on suure tõenäosusega lähedane otsitavale perioodile *r* vastava Fourier' sageduse *q* /r lambdandale harmoonilisele, st tüüpiliselt RegA_lugem on täisarv, mis ligikaudu võrdub $\lambda \cdot \frac{q}{r}$.

Miks ligikaudu?
(2 põhjust!)

Ehkki käesolevas simulatsioonis me juba graafikult näeme, mitmenda harmoonilise, st mitmenda piigi (λ = 0,1,2 või 3) juurde lugem sattus, siis tegeliku kvantarvuti puhul me ühestainsast lugemist temale vastavat lambdat ega ammugi siis ka perioodi kätte ei saa. **Katseid tuleb korrata**. Kahjuks tuleb silmas pidada, et peale väljalugemist kvantbit-registri olek on kokku kukkunud (loetud arvule RegA_lugem vastavaks omaolekuks) ja korduskatseid tuleb teostada iga kord **otsast peale**, st alates registrite algolekust.

On võimalik tõestada, et piisavalt ühelähedase tõenäosusega õige otsitava perioodi saamiseks on vajalik korduste arv $\text{ceil}(\ln(q)) = 6$ ja korduslugemitest on nn ahelmurrumeetodil väikese arvutusega tavalisel arvutil võimalik leida lamdad ja seega ettevalitud suuruse $q = 243$ kaudu ka otsitav periood. Meenutame, et "päris" koodimurdmisülesandes on *q* väga palju suurem arv, tal on vähemalt kaks korda niipalju kümnendkohti, kui tegureiks lahutataval arvul *n*.
Meie siin ahelmurrumeetodisse ei lasku, sest suurus *q* /r ja seega *r* on meil lihtsalt aimatav juba vähemast kui 6-st katsest. (Siinkohal unustame ära käesoleva simulaatori poolt võimaldatud jumaliku positsiooni, millelt vaadates graafikut ülal saame me ainsa lugemivõtuga või selletagi kätte suuruse *q* /r ja et periood *r* on meil koguni teada algusest peale --- **tegelik kvantarvuti seevastu ei näita meile ei muud kui lugemit**)

ceil(*x*) on reaalarvule *x* lähima suurema täisarvu leidmise funktsioon *MC*-s

TEE KORDUSKATSEID (sama astendusaluse x-ga)! Pane iga kord endale kirja lugem RegA_lugem ja määra neist otsitav periood *r* !

Kustkohast töölehel ülapool ja kuidas on õige simuleerida katse kordamist?

Perioodilise funktsiooni $f_{xamodn}(z)$ määratlusest §2 alguses tuleneb, et kuna $x^0 = 1$ ning alati 1-e jagamisel täisarvuga *n* on jäägiks 1 ja kuna $x^{0+r} = x^r$, siis

$\text{mod}\left(x^r, n\right) = 1$, kust saame kirjutada järgnevad võrdused:

$$\text{mod}\left[\left(x^{\frac{r}{2}}\right)^2, n\right] = 1 \rightarrow \text{mod}\left[\left(x^{\frac{r}{2}}\right)^2 - 1, n\right] = 0 \rightarrow \text{mod}\left[x^{\frac{r}{2}^2} - 1^2, n\right] = 0 \rightarrow$$

$$\rightarrow \text{mod}\left[\left(x^{\frac{r}{2}} - 1\right) \cdot \left(x^{\frac{r}{2}} + 1\right), n\right] = 0$$

Anna *r* -le leitud väärtus ja kontrolli (kopeerides mod-avaldise) arvuliselt nende võrduste õigsust!

Viimasest võrdusest järeldub, et kui juhtumisi üks täisarvudest korrutises (…-1)(…+1) tervikuna ei jagu *n*-ga, siis kindlasti vähemalt üks neist peab omama ühistegurit *n*-ga. Seega on meil hea shanss leida koodimurdmisülesandes otsitavad arvu *n* tegurid, kui me arvutame (lihtsa Eukleidese algoritmiga peast või tavalisel arvutil) suurimad ühisjagajad täisarvu *n* ja täisarvude $x^{\frac{r}{2}} + 1$ ning $x^{\frac{r}{2}} - 1$ vahel.

tavalisel arvutil) suurimad ühisjagajad täisarvu n ja täisarvude $x^{\frac{r}{2}}+1$ hiing $x^{\frac{r}{2}}-1$ vahel.

Antud juhul on periood $r = 4$, kusjuures $n = 15$ $x = 13$

$$x^{\frac{r}{2}} + 1 = 170 \qquad x^{\frac{r}{2}} - 1 = 168$$

Suurim ühisjagaja (greatest common divisor) on (vt ntx EE) sama, mis *suurim ühistegur.*

$$\gcd\left(x^{\frac{r}{2}} + 1, n\right) = 5 \qquad \gcd\left(x^{\frac{r}{2}} - 1, n\right) = 3$$

Siit paistavadki *n*-i tegurid.

Ülesanne on lahendatud ja *n* -i tundmatud tegurid leitud ning neil põhinev shiffer või turvakood samahästi kui lahti murtud!

Tõepoolest $n \text{ factor} \rightarrow 3 \cdot 5$.

☜ See üksainus *MC* operatsioon teeb siin ju ära kogu algoritmi töö, eks ole !? Tegurite leidmine pole *MC*-le probleem isegi veel ka 20-kohalise *n* puhul:

$$12345678987654321 \text{ factor} \rightarrow 3^4 \cdot 37^2 \cdot 333667^2$$

Saamaks ettekujutist ülesande eksponentsiaalselt kasvavast mahukusest, võiks siin faktoriseeritava arvu pikkust suurendada kümne koha kaupa.... Kelle arvuti tegurdab enne eksamit ära üle 100-kohalise korrutise kahest tundmatust algarvust, saab hindeks A ja selle arvuti ostan väga hea hinnaga ära...(-:

Probleemülesanne:
Shori algoritm võib ka eimidagi anda. Proovi simulatsiooni lasta joosta valikuga $x = 14$ (kui $n = 15$) või $x = 17$ (kui $n = 21$). Mille taha ülesande lahendamine kinni jääb?

---------- Töölehe lõpp ----------

Täiendatud - E aug 29 22:26:44 2011

Kvantbitis tekkivate vigade kvantarvutusliku korrigeerimise simulatsioon

Vähegi suurema hulga kvantbittide süsteemi oleku ülikiire lagunemine vaid tihedusmaatriksiga kirjelduvaks segaolekuks on füüsikaline paratamatus. Probleem on sedavõrd tõsine, et mõned autorid on pidanud kvantarvutamist igaveseks määratud olema vaid üks ilus teoreetiline distsipliin.

Olulise optimismilaine tekitas aga hiljuti avastatud võimalus kvantarvutil iseendas tekkivaid vigu kvantarvutuslike meetoditega parandada. Osutus, et ümbritseva maailma mõjutuste suhtes veakindla kvantbiti saamiseks tuleb kasutada minimaalselt +4-st abi-kvantbitist koosneva liitsüsteemi evolueerimist läbi paljulülilise "raudvara" (füüsikaliselt -- EM-väljad/laserimpulsid, matemaatiliselt -- unitaarsed operaatorid), mille käigus info kaitstava kvantbiti seisundi kohta kodeeritakse 5 kvantbiti põimseisundisse [R. Laflamme, *et al*, Phys. Rev. Lett. **77**, 198, 1996 ja viited sealsamas, vt ka lingid kursuse veebis].

Enne algoritmi simulatsiooni juurde asumist on vaja matem.kirjeldusse sisse tuua

1. Mudelvead ühe- ja mitme-kvantbitilises registris

2-Dim Hilberti ruumis kvantbiti **igasugune evolutsioon**, kaasa arvatud **vigade teke on kirjeldatav** 3-e **Pauli maatriksi** ja 2x2 ühikmaatriksi kaudu (üldjuhul nende lineaarkombinatsioonina):

$$\sigma_x := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \sigma_y := \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \sigma_z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad I := \text{identity}(2) \qquad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

1.1 Tagasipõige tava-kvantmehaanikasse

Pauli maatriksid kirjeldavad, muuseas, osakese (1/2-spinniga elektroni, aga ka footoni jms) polarisatsiooni vastaval füüsikaliste suuruste ja nende keskväärtuste kohta käivatele kvantmehaanika valemitele.

Kordame definitsioone failist **1_RegisterjaGate.mcd**

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad \langle 0| := |0\rangle^T \qquad \langle 0| = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

nn bra-vektor Diraci bra-ket tähistuses

$$U(\theta,\phi) := \begin{pmatrix} \cos(\theta) & \sin(\theta)\cdot\exp(-i\cdot\phi) \\ \sin(\theta)\cdot\exp(i\cdot\phi) & -\cos(\theta) \end{pmatrix}$$

Miks me tähistasime seda maatriksit tähega U ?

Nii näiteks on mikrofüüsikalise suuruse spinni ehk omapöörlemis-impulssmomendi vektori z-suunalist komponenti kirjeldav operaator σ_z (ühikutes 1/2 h*), mille omavektorid (füüsikalise ruumi pöörete suhtes spiinorid), vastavalt omaväärtustele +1 ja -1 (st +1/2 h* ja -1/2 h*), on :

$$\sigma_z\cdot|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \sigma_z\cdot|1\rangle = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$

kusjuures muidugi $-1\cdot\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$

ehk *MC* erifunktsioonidega sama tulemus:

$$\text{eigenvec}(\sigma_z, 1) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \text{eigenvec}(\sigma_z, -1) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Samas aga x-suunaline spinnikomponent (footoni puhul nn Stokes'i parameeter, mis iseloomustab polarisatsiooni 45-kraadise kallakusega komponenti), mis ei oma z-komponendiga korraga määratud väärtust, sest vastavad operaatorid ei kommuteeru:

$$\sigma_x\cdot\sigma_z - \sigma_z\cdot\sigma_x = \begin{pmatrix} 0 & -2 \\ 2 & 0 \end{pmatrix}$$

kuid mis muidugimõista, omab ka sama kahte omaväärtust +1 ja -1 , olles täpselt määratud enda omaolekuis

$$\text{eigenvec}(\sigma_x, 1) = \begin{pmatrix} 0.707 \\ 0.707 \end{pmatrix} \qquad \text{eigenvec}(\sigma_x, -1) = \begin{pmatrix} -0.707 \\ 0.707 \end{pmatrix}$$

mis, nagu peabki 2-seisundilise süsteemi kvantmehaanilises kirjelduses olema, on 45 kraadi pööratud algse suhtes:

$$|/\rangle := U\left(\frac{\pi}{4}, 0\right)\cdot|0\rangle \qquad |/\rangle = \begin{pmatrix} 0.707 \\ 0.707 \end{pmatrix}$$

Ekskursi lõpetuseks füüsikaliste suuruste kvantmehaanilisse kirjeldusse näitame viimaks, et selles olekus on spinni z-suunalise komponendi keskväärtus 0, sest mõlemad 2-st mõõtmistulemusest on võrdsed, vastandmärgilised ja võrdtõenäosed.

Täiendatud - E aug 29 22:26:44 2011

Kasutame mugavat Dirac'i bra|ket-tähistust, avaldades bra-vektori kui horisontaalseks **T**ransponeeritud ja kaaskomplekssseks tehtud (mis reaalsete komponentidega vektorite puhul tarbetu operatsioon) ket-vektori

$$\langle/| := \overline{\left(|/\rangle^T\right)} \qquad \langle/| = (0.707 \quad 0.707)$$

Kvant.meh. keskväärtuse leidmise valem annabki 0-i:

$$\langle/|\cdot\sigma_z\cdot|/\rangle = 0$$

Soovi korral saab siinkohal meenutada - neid n.-ö. elama pannes -- mitmeid põhimomente 2-seisundilise süsteemi kvantmehaanikast, kui asendada/täiendada valemit ülal teiste operaatorite ja bra-ket vektoritega.

Läheme tagasi (st edasi) kvantarvutite problemaatika juurde

1.2 Faasinihke-, ümberviske- (*bit flip*) ja faasinihkega ümberviske**viga** kvantbitis

Faasinihke-vea teke kirjeldub 3.-nda Pauli maatriksi toimega

$$\sigma_z \cdot \begin{pmatrix} komp_1 \\ komp_2 \end{pmatrix} \rightarrow \begin{pmatrix} komp_1 \\ -komp_2 \end{pmatrix}$$

ehk see = $komp_1 \cdot |0\rangle - komp_2 \cdot |1\rangle$

Vea nimetus tuleneb teise komponendi ette miinusmärgi tekkest (ehk pööre 180^0)

Ümberviske-vea teke kirjeldub 1.-e Pauli maatriksi toimega

$$\sigma_x \cdot \begin{pmatrix} komp_1 \\ komp_2 \end{pmatrix} \rightarrow \begin{pmatrix} komp_2 \\ komp_1 \end{pmatrix}$$

Nool --> tähendab, et tulemuse saamiseks on kasutatud *MC* sümbolarvutust

Faasinihkega ümberviske-vea teke kirjeldub 2.-e Pauli maatriksi toimega

$$\sigma_y \cdot \begin{pmatrix} komp_1 \\ komp_2 \end{pmatrix} \rightarrow \begin{pmatrix} -komp_2 \cdot i \\ komp_1 \cdot i \end{pmatrix}$$

Komponendid on vahetunud, suhteline faas on keeratud 180 kraadi ning absoluutne - 90 kraadi (aga olekuvektori komponentide absoluutne faas pole füüsikaliselt oluline)

Vea mitteteke kirjeldub identsusteisendusega ehk ühikmaatriksi toimega

Näita seda, täites sümbolarvutuse tühi (punane) platsihoidja vajalikuga ja klõpsates vaba pinda töölehel!

$$\blacksquare \cdot \begin{pmatrix} komp_1 \\ komp_2 \end{pmatrix} \rightarrow$$

Kokkuvõttes igasugune (kombineeritud) viga kirjeldub 2x2 maatriksiga, mis on nende 4 maatriksi lineaarne unitaarne superpositsioon.
Kuid seni on meil mudel vaid vigade kirjeldamiseks **ühes kvantbitis,** meil aga reeglina tegemist paljudest kvantbittidest koosneva kvantbittregistriga. Enne kui üldistada veatekke kirjeldust neile, tuleb meil sisse tuua mitmekvantbitilise liitsüsteemi kirjeldamiseks vajalik matemaatiline (ja simulatsiooni) aparaat.

1.3 Olekuvektorite ja operaatorite(maatriksite) otsekorrutis

Defineerime siin kollapseeritud alal (Area) vektorite kui ka maatriksite otsekorrutise operatsiooni, kirjutades selle (erinevalt otsekorrutisest esimesel töölehel) tavapärase matemaatilise kirjaviisi võimaldamiseks **tehtemärk-funktsioonina,** kus A ja B on **suvaliste ridade/veergude arvuga maatriksid**)

Demonstreerime äsjadefineeritud otsekorrutisetehte märgi kasutamist numbrilisel ja sümbolarvutamisel:

$$\begin{pmatrix}1\\0\end{pmatrix}\otimes\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}0\\1\\0\\0\end{pmatrix} \qquad \begin{pmatrix}a_1\\a_2\end{pmatrix}\otimes\begin{pmatrix}b_1\\b_2\end{pmatrix}\rightarrow\begin{pmatrix}a_1\cdot b_1\\a_1\cdot b_2\\a_2\cdot b_1\\a_2\cdot b_2\end{pmatrix}$$

Vaatame ka selliseid vektorite otsekorrutisi, mida meil küll edasises vaja pole:

$$\begin{pmatrix}a_1\\a_2\end{pmatrix}\otimes\begin{pmatrix}b_1 & b_2\end{pmatrix}\rightarrow\begin{pmatrix}a_1\cdot b_1 & a_1\cdot b_2\\a_2\cdot b_1 & a_2\cdot b_2\end{pmatrix} \qquad \begin{pmatrix}a_1 & a_2\end{pmatrix}\otimes\begin{pmatrix}b_1\\b_2\end{pmatrix}\rightarrow\begin{pmatrix}a_1\cdot b_1 & a_2\cdot b_1\\a_1\cdot b_2 & a_2\cdot b_2\end{pmatrix}$$

Kui sümbolarvutustehe punane, siis kasutatav *MC* versioon ei "ei oska" sümbolarvutust äsja defineeritud otsekorrutisoperaatori-funktsiooniga ja need otsekorrutise näited tuleb endal välja arvutada.

Võrdle saadud 2-ht viimast avaldist korrutistega tavalise maatriksite korrutamise eeskirja järgi, mis annavad, vastavalt, 2x2 maatriksi ja skaalari (vektorite skal.korrutise):

$$\begin{pmatrix}a_1\\a_2\end{pmatrix}\cdot\begin{pmatrix}b_1 & b_2\end{pmatrix}\rightarrow\begin{pmatrix}a_1\cdot b_1 & a_1\cdot b_2\\a_2\cdot b_1 & a_2\cdot b_2\end{pmatrix} \qquad \begin{pmatrix}a_1 & a_2\end{pmatrix}\cdot\begin{pmatrix}b_1\\b_2\end{pmatrix}\rightarrow a_1\cdot b_1+a_2\cdot b_2$$

Proovi otsekorrutist mitmesuguste 2x2 maatriksitega, mille elemendid numbrilised, kusjuures otsekorrutisemärk kopeeri (Ctrl+C ja Ctrl+V) eestpoolt. Enne avaldisele arvutamiseks võrdusmärgi panemist ennusta tulemus!

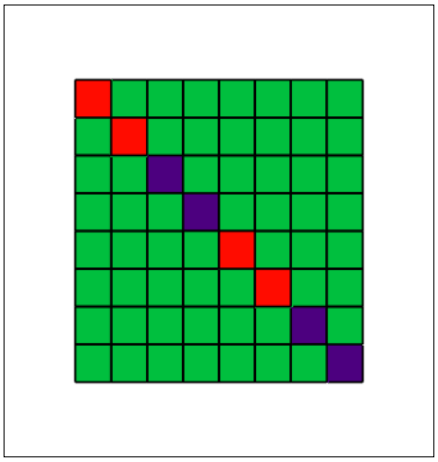
Nüüd on meil vajalikud vahendid olemas,
et kirjeldada/simuleerida

1.4 Faasinihke-, ümberviske- (*bit flip*) ja faasinihkega ümberviske**viga** kvantbitt-**registris**

Veaoperaatorite üldistuse **mitmebitilisele registrile** tagab otsekorrutustehe. Näiteks faasinihkeviga 2.-s kvantbitis 3-kv.b.-s registris tekitub operaatori mõjul (otsekorrutis on assotsiatiivne ja tegelikult sulge ei vajaks), mis esitub:

$$\mathrm{I}\otimes\left(\sigma_z\otimes\mathrm{I}\right)=\begin{pmatrix}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&-1&0&0&0&0&0\\0&0&0&-1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&-1&0\\0&0&0&0&0&0&0&-1\end{pmatrix}$$

ehk



0-d on rohelised

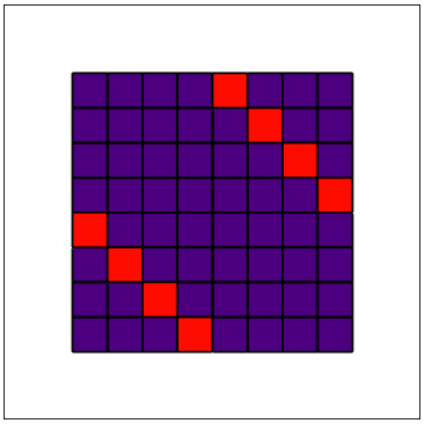
$\mathrm{I}\otimes\left(\sigma_z\otimes\mathrm{I}\right)$

Mis oli I ja miks σ on siin I-de vahel?

Näiteks ümberviskeviga 1.-s kvantbitis 3-kvantbitises registris tekitub sellise operaatori mõjul:

$$\sigma_x\otimes(\mathrm{I}\otimes\mathrm{I})=\begin{pmatrix}0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&1\\1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\end{pmatrix}$$

0-d on sinised



$\sigma_x\otimes(\mathrm{I}\otimes\mathrm{I})$

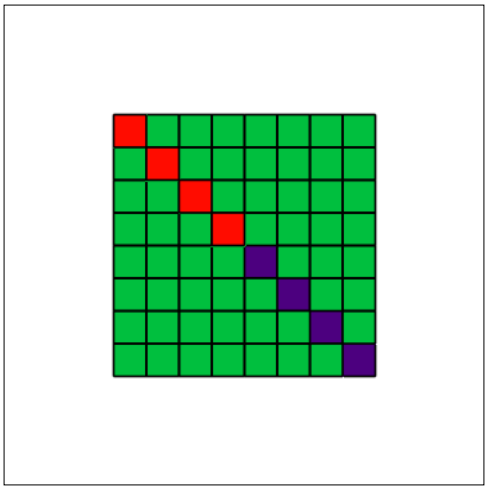
Kui anda 3-se registri üld(superpos)seisund, kus a-d tähistavad paralleelselt sissekirjutatud kümnendarvude 0....7 tõenäosuste amplituude, siis vaadates selle maatriksiga teisendust

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix} \rightarrow \begin{pmatrix} a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

näeme, et amplituudide ümberpaigutus on selline, mis vastab 3-ses registris |xxx> vanimat kahenddjärku kandva (vasakpoolseima) kvantbiti komponentide ümberviskele (vt punkt 1.2 ja veendu väite õigsuses)

Aga näiteks faasinihkeviga 1.-s kvantbitis 3-ses registris kirjeldub sellise operaatoriga:

$$\sigma_z \otimes (I \otimes I) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$



$\sigma_z \otimes (I \otimes I)$

Iseseisvaks uurimiseks :

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \cdot \rightarrow$$

Täites vasakul oleva sümbolarvutuse tühja platsihoidja vajalikuga, veendu,
et amplituudide märgimuutus on selline, mis vastab 3-ses registris |xxx> vanimat 2-ndjärku kandva (vasakpoolseima) kvantbiti faasinihkele (vt punkt 1.2)

Harjutus. Proovi ise eelneva eeskujul (ja kopeerides eelnevast töölehetükke) uurida faasinihkega ümberviske-vea teket 3-e kvantbitilise registri mingis kvantbitis (maatriksi pilte tee kaks - reaal- ja imaginaarosa jaoks eraldi) !

-------- Töölehe lõpp -------

29.08.2011 - viimati redigeeritud

Kvantbitis tekkivate vigade kvantarvutusliku korrigeerimise simulatsioon (järg e. osa B)

1. Kordame edasiseks vajalikud definitsioonid osast A:

2-Dim Hilberti ruumis kvantbiti igasugune evolutsioon, kaasa arvatud **vigade teke on kirjeldatav** 3-e Pauli maatriksi ja 2x2 ühikmaatriksi kaudu (üldjuhul nende lineaarkombinatsioonina):

$$\sigma_x := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \sigma_y := \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \sigma_z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad I := \text{identity}(2) \qquad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Defineerime ka siin vektorite ning maatriksite otsekorrutise operatsiooni $\otimes(A, B)$

Paremklahv ja "Expand" !

Asume nüüd asja kallale (NB! Vigade parandamine on mitte ainult kvantarvutis, vaid üldse... üks keeruline tegevus :-).

2. Laflamme' skeem korrigeerimaks ühte üldtüüpi viga 5 põimitud kvantbitiga registris

(Ava kõrvale ingliskeelne lülidiagramm, millel järgmised etapid)

1. Encode a given state, |psi>, as an entangled state of 5-qubits
2. Simulate the introduction of an error in the entanglement
3. Decode the (buggy) state of the entangled qubits
4. Determine the "error syndrome"
5. Given the error syndrome, apply the appropriate unitary operator (rotation) to correct the state.

2.1 5-kvantbiti-operaatorid ja sisendseisundi kodeerimine sõlmseisundiks

Kõigepealt on meil simulatsiooniks vaja juurde defineerida operaatoreid, mis moodustaksid skeemi lülid.

Definerime abioperaatori $\Uparrow$(n,N), mis viib n-nda kvantbiti N-bitises registris "0"-seisundist "1"-seisundisse. See avaldub otsekorrutistena 2x2-maatriksitest, kus vastava järjenumbriga n ühikmaatriks on asendatud fermioni sünnioperaatoriga (ehk operaatoriga, mille toime kvantbitile avaldub:

$$\begin{pmatrix}0&0\\1&0\end{pmatrix}\cdot|0\rangle=\begin{pmatrix}0\\1\end{pmatrix}$$, kuid $$\begin{pmatrix}0&0\\1&0\end{pmatrix}\cdot|1\rangle=\begin{pmatrix}0\\0\end{pmatrix}$$

st. "1"-seisundist kõrgemat seisundit lihtsalt pole ! (vastab füüsikaliselt Pauli keelureeglile: ühes seisundis saab olla vaid 0 või 1 fermioni)

```
⇑(mitmesQbitt, koguArvust) :=
    error("argumendid vigased -- järjenr. suurem koguarvust!")  if  mitmesQbitt > koguArvust
    error("argumendid vigased -- järjenr.<1")  if  mitmesQbitt < 1
    j ← koguArvust − 1
    M ← ⎛0  0⎞  if  mitmesQbitt = koguArvust
        ⎝1  0⎠
    otherwise
        M ← identity(2)
        while  j > mitmesQbitt
            M ← identity(2) ⊗ M
            j ← j − 1
        M ← ⎛0  0⎞ ⊗ M
            ⎝1  0⎠
        j ← j − 1
    while  j > 0
        M ← identity(2) ⊗ M
        j ← j − 1
    M
```

Näiteks, 3-bitises registris 2.-st kvantbitti "üles tõstev" operaator avaldub:

$$\Uparrow(2,3)=\begin{pmatrix}0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\end{pmatrix}$$

Asenda 1. -se argumendi väärtus "2" väärtustega
1 või 3 ja uuri töölehe klõpsamise järel saadavat tulemust!

Uuri selle operaatori toimet
3-kvantbitise registri olekule,
asendades avaldises lähtevektori
0-st erineva komponendi paiknemist
nii, et registri olek vastaks
kümnendarvudele 1, 2, 3,.....7

$$\Uparrow(2,3)\cdot\begin{pmatrix}1\\0\\0\\0\\0\\0\\0\\0\end{pmatrix}=\begin{pmatrix}0\\0\\1\\0\\0\\0\\0\\0\end{pmatrix}$$

Analoogiliselt saab defineeride maatriksii $\Downarrow$(n,N), mis viib n-nda kvantbiti N-bitises registris "1"-seisundist "0"-seisundisse.
See avaldub otsekorrutistena 2x2-maatriksitest, kus vastava järjenumbriga n ühikmaatriks on asendatud fermioni surmaoperaatoriga (ehk operaatoriga, mille toime kvantbitile avaldub:

$$\begin{pmatrix}0&1\\0&0\end{pmatrix}\cdot|1\rangle=\begin{pmatrix}1\\0\end{pmatrix}$$, kuid $$\begin{pmatrix}0&1\\0&0\end{pmatrix}\cdot|0\rangle=\begin{pmatrix}0\\0\end{pmatrix}$$ st. "vaakum-seisundist" |0> madalamat seisundit lihtsalt pole !

Meil aga pole mõtet mäluruumi sellele raisata, sest nagu teada kvantmehaanikast , ergastuse tekke- ja kao(surma/annihilatsiooni) operaator avalduvad vastastikku üksteisest transponeerimise ja kaaskompleksi võtmise teel

$$\begin{pmatrix}0&1\\0&0\end{pmatrix}^{T}=\begin{pmatrix}0&0\\1&0\end{pmatrix}$$ antud juhul kaaskompleksi võtta pole vajadust.

Sama transponeerituse-seos kehtib ka maatriksite $\Downarrow$(n,N) ja $\Uparrow$(n,N) vahel, sest ühikmaatriksi transponeerimine viimast ei muuda, aga otsekorrutise tehte ja transponeerimise tehte järjekord on vahetatav

Nüüd on meil vajalik olemas, et avaldada kodeerimisskeemi põhilüli -- 5- kvantbitiga opereeriva 'controlled-NOT' -lüli maatriks.
Kasutades asjaolu, et loogiliselt toimelt on see lüli identne kaheväljundilise 'XOR' -lüliga, kasutamegi tähistust 'XOR' defineeritavale maatriks-funktsioonile.
Maatriks-funktsiooni argumentideks on: CNOT-kontroll(kvant)biti järjenumber, kontrollitava biti järjenumber ja kvantbittide koguarv registris vms.

Et XOR avaldub just sellise valemiga sünni/surmaoperaatorite kaudu, ei oma sügavat füüsikalist tähendust.

$$\mathrm{XOR}(\mathrm{Ctrlbitt},\mathrm{FlipibMdat},\mathrm{Nst}):=\begin{bmatrix}\Uparrow(\mathrm{Ctrlbitt},\mathrm{Nst})\cdot\Uparrow(\mathrm{Ctrlbitt},\mathrm{Nst})^{T}\cdot\left(\Uparrow(\mathrm{FlipibMdat},\mathrm{Nst})^{T}+\Uparrow(\mathrm{FlipibMdat},\mathrm{Nst})\right)\ ...\\+\Uparrow(\mathrm{Ctrlbitt},\mathrm{Nst})^{T}\cdot\Uparrow(\mathrm{Ctrlbitt},\mathrm{Nst})\end{bmatrix}$$

Funktsioonidefinitsioonis on muutujaile pandud nimed selliselt, et nad vihjaksid muutuja tähendusele.

Veendume äsjadefineeritu õigsuses -- meile **esimesest töölehest tuttava 2-kvantbitises liitsüsteemis opereeriva CNOT-lüli** maatriksi saame, kui paneme sisse vastavad parameetrid:

$$\mathrm{XOR}(1,2,2) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Kodeerimisskeemi kohaselt on vajalik veel üks lüli -- 1 biti 45-kraadi võrra (ehk st spinni 90 kraadi võrra) pööraja R liitsüsteemis:
(kes ei tunne eriti huvi programeerimise vastu, ei pea süvenema funktsioonide progr.koodi - see ei ole obligatoorne mõistmaks lüli mõtet)

Funktsioonidefinitsioonis on muutujaile pandud nimed selliselt, et nad vihjaksid muutuja tähendusele.

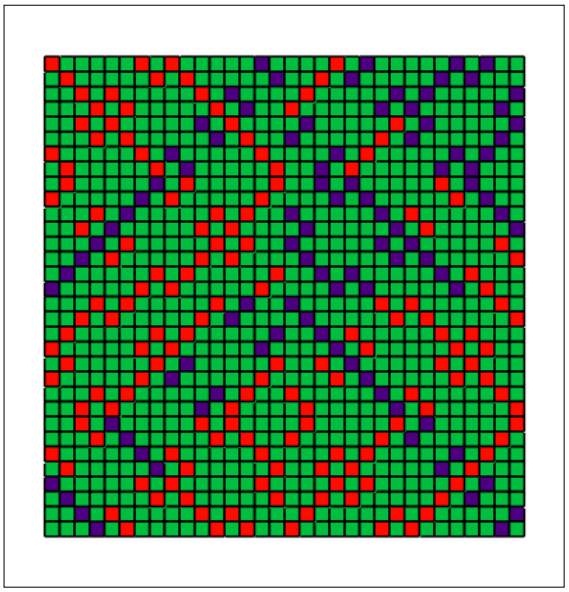
```
R(mitmesQbitt, koguArvust) :=
    error("argumendid vigased -- järjenr. suurem koguarvust!")  if  mitmesQbitt > koguArvust
    error("argumendid vigased -- järjenr.<1")  if  mitmesQbitt < 1
    j ← koguArvust − 1
    M ← [[1/√2, −1/√2], [1/√2, 1/√2]]  if  mitmesQbitt = koguArvust
    otherwise
        M ← identity(2)
        while  j > mitmesQbitt
            M ← identity(2) ⊗ M
            j ← j − 1
        M ← [[1/√2, −1/√2], [1/√2, 1/√2]] ⊗ M
        j ← j − 1
    while  j > 0
        M ← identity(2) ⊗ M
        j ← j − 1
    M
```

Nüüd on meil kõik olemas, et saavutada simulaatori selle alaetapi eesmärk -- leida kogu kodeerimisoperaator maatrikskorrutisena vastavalt lülide järgnevusele skeemis (vt. xerokoopia). Anname sellele maatriksile nimeks *Lencode* **.**

Lencode := XOR(1,5,5)·XOR(1,4,5)·R(1,5)·XOR(1,2,5)·XOR(1,3,5)·XOR(2,5,5)·R(2,5)·XOR(2,3,5)·XOR(2,4,5)·R(2,5

Kuidas on *Lencode* saamiseks korrutatud 10 maatriksit vastavuses sümbolitega lüliskeemis ?
(NB! ajas esimesena rakenduv opraatorilüli on otsekorrutises viimane st. **paremalt** esimene)

Siin ta on -- pildina



ja tabelina (hiireklõps tabelil lubab vaadata sisu)

Lencode =

	1	2	3	4	5	6
1	0.354	0	0	0	0	
2	0	0.354	0	0	0	
3	0	0	0.354	0	0.354	
4	0	0	0	0.354	0	0.
5	0	0	0.354	0	0.354	
6	0	0	0	0.354	0	0.
7	0.354	0	0	0	0	
8	0	0.354	0	0	0	
9	0	0.354	0	0	0	
10	0.354	0	0	0	0	
11	0	0	0	0.354	0	-0.
12	0	0	0.354	0	-0.354	
13	0	0	0	-0.354	0	0.
14	0	0	-0.354	0	0.354	
15	0	-0.354	0	0	0	
16	-0.354	0	0	0	0	

Lencode

$$\frac{1}{(\sqrt{2})^3} = 0.354$$

Võrdlustabel 5-kvantbitise registri baasivektorite tähistustele

Näidatud iga kv.biti seisund viiest ↓

Näidatud baasivektori No. ja hoitav 10-ndarv (kui teised komp.-d kõik =0)

Nii kodeerib see operaator kaitstava kvantbiti, kui see on seisundis |0>, kokku põimseisundiks
4 abibitiga (need alati algselt seisundis |0>)

Lencode·[|0> ⊗ [|0> ⊗ [|0> ⊗ (|0> ⊗ |0>)]]] =

	1
1	0.354
2	0
3	0
4	0
5	0
6	0
7	0.354
8	0
9	0
10	0.354
11	0
12	0
13	0
14	0
15	0
16	...

Klõpsa tabelil kerimisnuppude tekitamiseks!

Nii aga kodeerib see operaator kaitstava kvantbiti |1> -seisundis kokku põimseisundiks
4 abibitiga (need alati algselt |0>-seisundis)

Lencode·[|1> ⊗ [|0> ⊗ [|0> ⊗ (|0> ⊗ |0>)]]] =

	1
1	0
2	0
3	0
4	0.354
5	0
6	-0.354
7	0
8	0
9	0
10	0
11	-0.354
12	0

		null
|00000>	1	1
|00001>	2	2
|00010>	3	3
|00011>	4	4
|00100>	5	5
|00101>	6	6
|00110>	7	7
|00111>	8	8
|01000>	9	9
|01001>	10	10
|01010>	11	11
|01011>	12	12
|01100>	13	13
|01101>	14	14
|01110>	15	15
|01111>	16	16
|10000>	17	17
|10001>	18	18
|10010>	19	19
|10011>	20	20
|10100>	21	21
|10101>	22	22
|10110>	23	23
|10111>	24	24
|11000>	25	25
|11001>	26	26
|11010>	27	27
|11011>	28	28
|11100>	29	29
|11101>	30	30
|11110>	31	31
|11111>	32	32

13	-0.354
14	0
15	0
16	...

Anname nüüd ette simulaatorile kaitstava kvantbiti **suvalises olekus**, sisestades kaks nurka, mis määravad vektori orientatsiooni ühiksfääril (Poincare' sfääril footonite polarisatsiooni puhul ehk Bloch'i sfääril 1/2-spinniga osakeste või 2-seisundiliste ioonide/molekulide/vms korral)

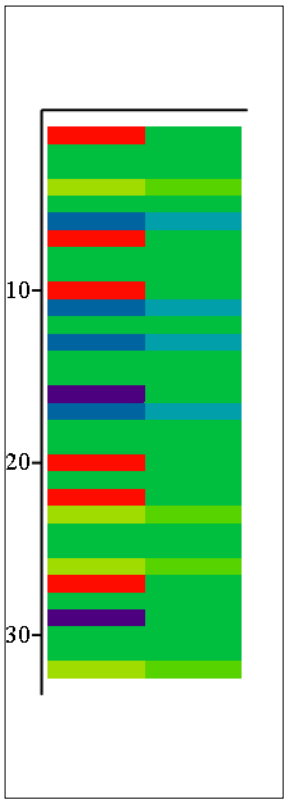
Muuda neid hiljem mõtestatult!

$\theta := 59 \cdot deg$ $\phi := 35 \cdot deg$ $Qubit := \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) \cdot \exp(i \cdot \phi) \end{pmatrix}$

5-e kvantbiti seisund (olekuvektor), mis kohe varsti n.-ö. **siseneb veatekketsooni** (vt skeem xerolehel!), avaldub:

$Kd := Lencode \cdot [Qubit \otimes [|0\rangle \otimes [|0\rangle \otimes (|0\rangle \otimes |0\rangle)]]]$

Vaatame (meil on see võimalik, kuna oleme simulatsioonile tänu paremas staatuses kui vanajumal reaalse kvantsüsteemi suhtes) seda põimseisundit kahes erinevas tema (üldjuhul kompleksarvuliste) komponentide esituses:



augment(Re(Kd), Im(Kd))

Kd =

	1
1	0.308
2	0
3	0
4	0.143+0.1i
5	0
6	-0.143-0.1i
7	0.308
8	0
9	0
10	0.308
11	-0.143-0.1i
12	0
13	-0.143-0.1i
14	0
15	0
16	...

Tabelis (klõpsa!) näidatavate komplekssete vektorikomponentide piltlikustamiseks on vasakul toodud 32x2 maatriks, mille vasak veerg on Re(Kd) ja parem veerg Im(Kd)

Muuda kaitstava kvantbiti algolekut tema olekuvektori nurkade muutmise kaudu (ekraani jagu eespool) ja uuri vektoris Kd tema kokkupõimimist abikavntbittidega

2.2 Ühe üldtüüpi vea tekke simulatsioon

Tekitame ühe vea, näiteks 4-ndas kvantbitis faasinihkega ümberviske, mida kirjeldab vektori Kd teisendus:

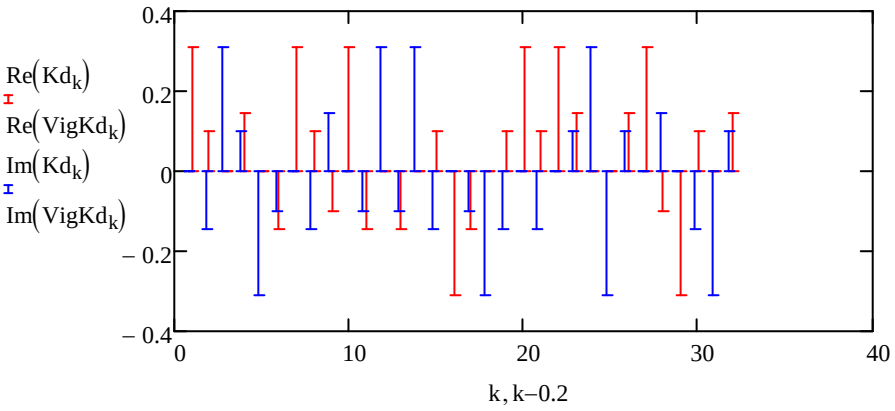
$$\mathrm{VigKd} := \left[\mathrm{I} \otimes \left[\mathrm{I} \otimes \left[\mathrm{I} \otimes \left(\sigma_y \otimes \mathrm{I}\right)\right]\right]\right] \cdot \mathrm{Kd}$$

Eriti selgelt on näha vea mõju paljudele vektori komponentidele vea graafilises esituses, kus kriipsud näitavad erinevust vektori rikkumata ja rikutud (*buggy)* komponentide vahel

k := 1 .. 32

VigKd =

	1
1	0
2	0.1-0.143i
3	0.308i
4	0
5	-0.308i
6	0
7	0
8	0.1-0.143i
9	-0.1+0.143i
10	0
11	0
12	0.308i
13	0
14	0.308i
15	0.1-0.143i
16	...



Nihe 0.2 eraldamaks siniseid punastest

Miks on graafikul kahte värvi kriipse?

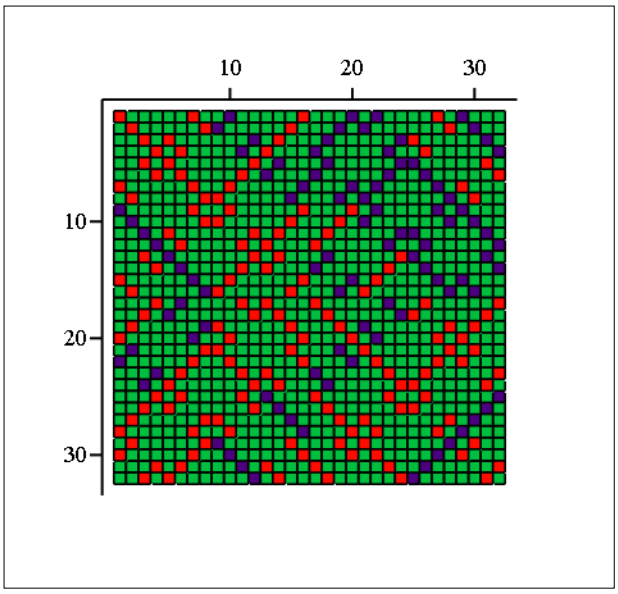
2.3 Vigase sõlmseisundi dekodeerimine

Nüüd peame laskma vigaseks muutunud kvantbitid läbi skeemi parema poole, mis on peegelpilt esimesest; selleks tuleb meile kasutada dekodeerimisoperaatorit *Ldecode* kujul

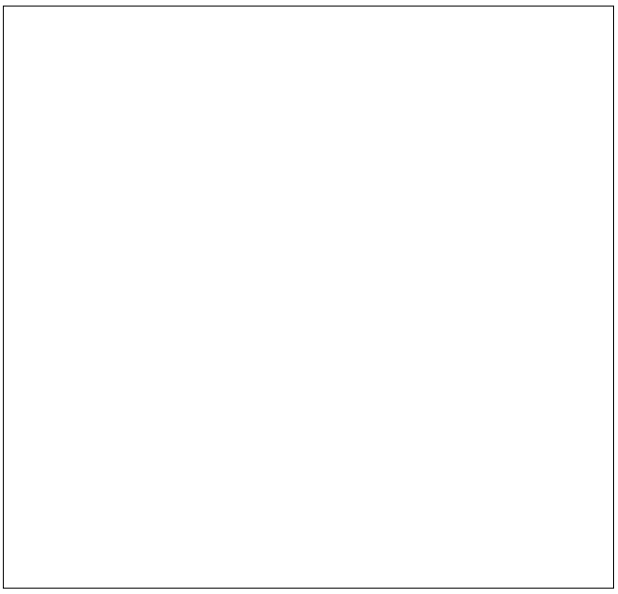
$$\text{Ldecode} := R(2,5)^{T}\cdot XOR(2,4,5)\cdot XOR(2,3,5)\cdot R(2,5)\cdot XOR(2,5,5)\cdot XOR(1,3,5)\cdot XOR(1,2,5)\cdot R(1,5)\cdot XOR(1,4,5)\cdot XOR(1,5$$

(NB! ajas esimesena rakenduv opraatorilüli on korrutises **paremalt** esimene)

Võrdluseks -- kodeerimismaatriks oli :



Ldecode



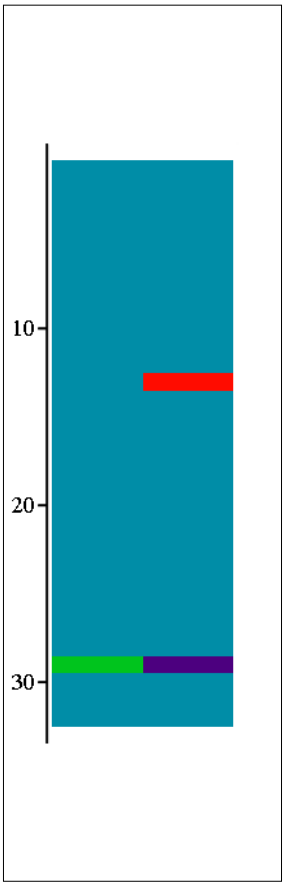
Lencode

Lüliskeemist väljub 5-bitine register seisundis: $\mathrm{VigOut} := \mathrm{Ldecode} \cdot \mathrm{VigKd}$,

mille tabel- ja piltesitus näitab,
et seisund on lahti põimitud kvantbittide
otsekorrutiseks, st **ei ole enam põimolek**:

$\mathrm{VigOut} =$

	1
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0.87i
14	0
15	0
16	...



$\mathrm{augment}(\mathrm{Re}(\mathrm{VigOut}), \mathrm{Im}(\mathrm{VigOut}))$

Huvi pärast kontrollime vektori pikkuse
ruudu normeeritust 1-le

$\left(\overline{\mathrm{VigOut}}\right)^{\mathrm{T}} \cdot \mathrm{VigOut} = 1$

Defineerime komponentide järjekorranumbrite 1...32 veeru:

$K_k := k$

Moodustame edaspidiseks mugavamaks esituseks sobiva
tabeli (see **pole** füüsikaline operaator!)
vektori komponentidest ja vastavatest mõõtmistulemuste
tõenäosustest:

$\mathrm{MTbl} := \mathrm{augment}\left[K, \mathrm{augment}\left[\mathrm{VigOut}, \left[\overrightarrow{\left(\overline{\mathrm{VigOut}}\right) \cdot \mathrm{VigOut}}\right]\right]\right]$

$\mathrm{MTbl} =$

	1	2	3
1	1	0	0
2	2	0	0
3	3	0	0
4	4	0	0
5	5	0	0
6	6	0	0
7	7	0	0
8	8	0	0
9	9	0	0
10	10	0	0
11	11	0	0
12	12	0	0
13	13	0.87i	0.758
14	14	0	0

15	15	0	0
16	16	0	...

2. veerg -- komp.

3. veerg -- selle mooduli ruut (tõenäosus)

Mugavuse huvides sorteerime äsjamoodustatud olekuvektori komponentide tabeli oodatavate mõõtmistulemuste tõenäosuste kahanevas järjekorras:

$$MTbl := reverse(csort(MTbl, 3))$$

MC erifunktsioonid - vt *Help*

$MTbl =$

	1	2	3
1	13	0.87i	0.758
2	29	0.282-0.403i	0.242
3	18	0	0
4	2	0	0
5	14	0	0
6	28	0	0
7	27	0	0
8	26	0	0
9	25	0	0
10	24	0	0
11	23	0	0
12	22	0	0
13	21	0	0
14	20	0	0
15	19	0	0
16	9	0	...

2.4 Abi-kvantbittide "mahamõõtmine" ja "veasündroomi" määramine

Kuna väljundseisund on otsekorrutisseisund, siis abi-kvantbittide mõõtmine nende seisundit ei muuda ja mingit tõenäosuslikku "kokkukukkumist" mõõtmisel ei toimu.

Kontrollime, kas väljundvektor faktoriseerub vanima (vasakpoolseima) kvantbiti otsekorrutiseks 4 noorema kvantbiti omaolekute otsekorrutisega, nagu peab Laflamme skeemi puhul olema. Juhul kui **ja**, siis kontrollfunktsioon väljastab (maatriksi M2x2 kaudu) vanima kvantbiti olekuvektori koos 4 noorema (abi)biti omaolekute kombinatsiooni ehk nn veasündroomi järjenumbiga: Mcheck(M32x3)

Siin on kollapseeritult MC vahenditega veasündroomi määraja

Selle funktsiooni toimel saame:

$$\text{Mcheck}(\text{MTbl}) = \begin{pmatrix} 13 & 0.87\text{i} \\ 29 & 0.282 - 0.403\text{i} \end{pmatrix}$$

Selle maatriksi parempoolne ehk nr. <2> veerg on vanima kvantbiti olekuvektor ja esimese rea esimene element on 4 noorema biti omaolekute kombinatsiooni ehk nn veasündroomi järjenumber

Eraldame vanima kvantbiti olekuvektori

$$\text{BuggyQubit} := \text{Mcheck}(\text{MTbl})^{\langle 2 \rangle} \quad ,$$

mis tuleb järgmises -- veakorrigeerimis-skeemi lõppetapis sellisest polarisaatorist läbi lasta (vektorit H.ruumis pöörata), millele vastav operaator 2x2 maatriksina sõltub nel- ja abi-kvantbiti mõõtmisjärgsest seisundist **vastavalt Laflamme kodeeringu jaoks välja töötatud Tabelile ---------->**

ErrorSyndrome2ActionTable :=

Syndrome |	Error Type* |	Action to correct
"|0000>"	"P5"	σ_z
"|0001>"	"BP5"	$\sigma_z \cdot (-i)$
"|0010>"	"B2"	$\sigma_y \cdot i$
"|0011>"	"P3"	σ_x
"|0100>"	"BP2"	$\sigma_x \cdot (-i)$
"|0101>"	"BP1"	$I \cdot i$
"|0110>"	"P1"	σ_z
"|0111>"	"BP3"	$\sigma_x \cdot (-i)$
"|1000>"	"B1"	σ_x
"|1001>"	"B4"	$\sigma_y \cdot i$
"|1010>"	"B5"	$\sigma_y \cdot i$
"|1011>"	"No Error"	$\sigma_y \cdot i$
"|1100>"	"BP4"	$\sigma_z \cdot (-i)$
"|1101>"	"P2"	σ_x
"|1110>"	"P4"	σ_z
"|1111>"	"B3"	$\sigma_y \cdot i$

*) Tabelis on kasutatud lühendeid, näiteks:
P5 on *Phase Shift Error* 5.ndas kvantbitis
B2 on *Bit Flip Error* 2.-s kvantbitis
BP1 on *Bit Flip and Phase Shift Error* 1.-s ehk kogu skeemiga kaitstavas kvantbitis.
(vt vigade mudelid eespool sellel töölehel)

Mäluvärskendus →/

2.5 Vea korrigeerimine -- kaitstava kvantbiti õigesse algolekusse tagasipööramine

Mõõtmise tagajärjel kukkusid (õigemini -- dekodeerijas juba pöördusidki) meil seega siis abibitid veasündroomi näitavasse omaolekute kombinatsiooni ja vanima (kaitstava) biti seisund eraldus otsekorrutisest välja sellisena:

$$\text{BuggyQubit} = \begin{pmatrix} 0.87\text{i} \\ 0.282 - 0.403\text{i} \end{pmatrix}$$

mis tuleb nüüd skeemi lõppetapis sellisest polarisaatorist läbi lasta (vektorit H.ruumis pöörata), millele vastav operaator 2x2 maatriksina sõltub nelja abi-kvantbiti mõõtmisjärgsest seisundist vastavalt Tabelile.

Võtame Tabelis 3.-st veerust veasündroomi järjenumbrile vastavast reast maatriksi paranduspöörde jaoks:

$$\text{RotatePlrstn} := \text{ErrorSyndrome2ActionTable}_{\text{Mcheck(MTbl)}_{1,1},3}$$

Nelja abi-kvantbiti mõõtmise tulemusena läksid need tõenäosusega 1 seisundisse - ehk teiste sõnadega -- saime teada "veasündroomi" , mis on:

$$\text{ErrorSyndrome2ActionTable}_{\text{Mcheck(MTbl)}_{1,1},1} = \text{"|1100>"}$$

ja veatüüp on:

$$\text{ErrorSyndrome2ActionTable}_{\text{Mcheck(MTbl)}_{1,1},2} = \text{"BP4"}$$

Esimene/vanim kvantbit on mingis superpositsioonilises (vigases/vales) olekus, mis reaalsuses on meile teadmata nagu tema esialgnegi olek, mida aga sellest hoolimata saame nüüd taastada korrigeeriva pööramisoperatsiooniga --- polarisaatorile vastava maatriksi ja pööratava vektori (maatriks)korrutamisega, seega --

veakorrigeerimise lõppetapp on paranduspööre:

$$\text{RestoredQubit} := \text{RotatePlrstn} \cdot \text{BuggyQubit}$$

Võrdle kvantbiti taastatud seisundit (i) algse ja (ii) vigasega:

$$\text{RestoredQubit} = \begin{pmatrix} 0.87 \\ 0.403 + 0.282\text{i} \end{pmatrix}$$

(i) $$\text{Qubit} = \begin{pmatrix} 0.87 \\ 0.403 + 0.282\text{i} \end{pmatrix}$$

(ii) $$\text{BuggyQubit} = \begin{pmatrix} 0.87\text{i} \\ 0.282 - 0.403\text{i} \end{pmatrix}$$

Uurimaks kogu algoritmi tööd simulaatoril, muuda ülal kvantbiti algseisundit, veatüüpi ja kvantbiti numbrit, milles viga tekib.

Harjutus *for advanced users:* kas suudad korrigeerimisskeemi "üle mängida", st saavutada füüsikaliselt mõtestatult olukord, kus enam ei taastu kaitstava biti seisund?

Küsimusi enesekontrolliks

1. Mitmedimensionaalses ruumis töötab operaator *Ldecode* ?

2. Dekodeerija väljundolekut uurides saime 4 arvu: $\mathrm{Mcheck(MTbl)} = \begin{pmatrix} 13 & 0.87\mathrm{i} \\ 29 & 0.282 - 0.403\mathrm{i} \end{pmatrix}$. Milliste neist väärtusi saame tegelikus kvantarvutis ja millised jäävad teadmata?

3. Kas protokoll parandab vaid vea, mis tekkis kaitstavas kvantbitis?

4. Kas protokoll aitab vea tekke vastu rohkem kui ühes kvantbitis?

5. Kas protokoll aitab rohkem kui ühe vea tekke vastu ühes kvantbitis?

6. Miks dekodeerija annab välja otsekorrutisseisundi (kui kodeerija sisendis oli otsekorrutisseisund) ja seda isegi siis, kui vahepeal tekib viga ühes kvantbitis?

7. Miks protokoll ei näe ette kaitstava kvantbiti mõõtmist?

---------- Töölehe lõpp ----------

Kvantkrüptograafiast

Sisukord

Täiendatud - R nov 13 17:38:49 2009

1. Sissejuhatus

Meenutame, et avaliku võtmega RSA krüptosüsteemi turvalisus põhines tõigal, et pole olemas piisavalt kiiret arvutusalgoritmi suurte arvude tegureiks lahutamiseks. Rõhutame, et vähemasti tänaseni **pole matemaatika tõestanud**, et niisugust algoritmi (klassikalisele arvutile) on võimatu luua.

Kui luuakse kvantarvuti, mis realiseerib Shori algoritmi, siis polegi tõestus enam praktiliselt oluline – sellega on kvantfüüsika nagunii põhja lasknud RSA kui tänapäevase krüptograafia ühe lipulaeva.

Samas kvantfüüsika teiselt poolt pakub – või täpsemini, on eksperimentaalselt kontrollituna ja kommertsialiseerimisvalmina juba pakkunud – asemele põhimõtteliselt pealtkuulamiskindla salasideliini. Rõhutame, et kvantkrüptograafilise sideliini turvalisus põhineb kvantmehaanika alustõdedel, mille paikapidavust **on tõestanud kogu tänapäevane füüsika** – nii teooriad kui tohutu hulk katseid ja praktilisi tehnoloogilisi rakendusi.

Kui makroskoopilisi infokandjaid, näiteks paberilehte või elektriimpulsse kaablis, saab lugeda ilma et nendega midagi juhtuks, siis kvantmehaanilise objekti "lugemine" , ükskõik kui "delikaatselt" seda püütaks teha, paratamatult muudab objekti olekut. Veelgi enam, footoni puhul lõpeb iga diagnostikaprotseduur footoni äraneeldumisega detektoris, kui just ei kasutata erimeetodeid/tehnikaid. Seega on Alice'il ja Bobil kerge avastada Eve sekkumist kvant-sidekanalisse. Eve'i ei aita ka idee püüda mitte torkida (kvantmehaanilise mõõtmise mõttes) sideliini pidi liikuvaid kvantobjekte, vaid kopeerida nende olekud üle omastkäest võetud kvantosakestele ja uurida/mõõta/lugeda siis neid. Sest see idee ei tööta, kuna kehtib teoreem: kvantolek pole kloonitav.

Kvantkrüptograafiale on iseloomulik vajadus kahe kanali järele: lisaks kvantkanalile peab olema klassikaline kanal – tavaline sideliin või avalik eeter. Klassikalise infokanali põhimõttelist vajalikkust on kerge mõista, kui meenutame sissejuhatava loengu animatsiooni Einsteinile "õudsest kaugmõjust" põimolekus osakeste vahel. "Õudne kaugmõju" levib küll hetkeliselt, kuid ei võimalda superluminaalset infoedastust. Informatsiooni kannab "õudne kaugmõju" vaid koos klassikalise infokanaliga, mis aga tagabki piirkiiruse *c* mitteületatavuse.

Käesoleval töölehel tutvume – peale ettevalmistavat materjali – kvantkrüptograafilise sideliini kahe erineva tüübiga: (i) üksikute osakeste ja (ii) põimolekus osakestepaaride saatmisel põhinevaga. Mõlema tüübi mitmetest füüsikalistest realiseerimisvõimalustest vaatleme konkreetselt footonite polarisatsiooni muutmisel ja mõõtmisel põhinevaid.

Nagu varasemategi kvantarvuti-töölehtede puhul, on oluline silmas pidada järgmist. Klassikalist krüptograafiat käsitlenud töölehel oli praktiliselt kasutatav "päris elus" krüpteerimismasin ja RSA-süsteemi tutvustavat osa nimetasime simulaatoriks vaid selles mõttes, et ehkki kõik töötas nagu "päris", polnud valitud lähtearvud piisavalt suured turvalisuse tagamiseks. Seevastu kvantsidekanali tööd – peale kirjeldamise muidugi – saab arvutil "elusana" üksnes **simuleerida** selle sõna kõige vängemas tähenduses, sest arvuti **ei käitu** kvantmehaanilise objektina.

2. Eeldusi

Eeldame, et edastatav teade on mingitmoodi juba binaarseks kodeeritud, st **kujutab endast "1"-de ja "0"-de jada**. See ei pruugugi olla salateate enda kodeering, vaid näiteks Vernami šifri võti. Tegelikult ongi kvantkrüptograafiat oluliselt lihtsam realiseerida just võtmete edastamiseks, sest pole probleemi osa sümbolite kaotsiminekust Eve tegutsemise või näiteks footonite kaablis neeldumise tagajärjel. Lihtsalt tuleb Alice'l ja Bobil leppida kokku kasutada võtit sellisena nagu ta Bobile kohale jõudis. Siit sugeneb lisa-turvamoment, sest ajal, mil Eve pealt kuulab, pole võtit veel defineeritudki!

Kuna ka RSA-süsteemi võb kasutada muus süsteemis šifreeringu võtme turvaliseks edastamiseks, võib kvantkrüptograafiat käsitada kui alternatiivi RSA-le (päästvat alternatiivi, kui Shori algoritm praktiliselt tööle hakkab !). Seetõttu kasutatakse kvantkrüptograafia tähenduses inglise keeles ka terminit *quantum key distribution* versus *public key distribution* (RSA).

Eeldame, et kvantkanali kõrval vajalik klassikaline kanal on avalik selles mõttes, et igaüks võib seda vabalt pealt kuulata, kuid siiski **keegi ei saa moonutada** sedapidi liikuvat infot. Küllalt hästi vastab niisugusele nõudele näiteks raadioside.

Eeldame, et kvantkanalisse suunatav piisavalt nõrk valgus koosneb **üksikutest footonitest.** Vastupidi esmasele arvamusele pole seda nõuet sugugi kerge täita (kõrge statistilise usaldatavusega).

3. "Kvantraha" ja mitteklooni tavuse teoreem
"Quantum Money" and No-Cloning Theorem

Eve'i saaks kopeerida sideliini pidi liikuvate kvantobjektide olekuid üle omastkäest võetud kvant-osakestele ja uurida/mõõta/lugeda siis neid omaette, ilma et Alice ja Bob saaks pealtkuulamisest aimu. See oleks võimalik, kui oleks võimalik kloonida kvantolekut nagu on põhimõtteliselt võimalik valmistada täiuslikke koopiaid rahatähtedest. Kvantobjektide puhul see aga pole võimalik, mida hästi väljendab S.Wiesneri poolt 1983.a. publitseeritud mittevõltsitava "kvantraha" idee.

Käsitleme kahte 1-le normeeritud **erinevat** olekuvektorit |a> ja |b> , mis **pole ortogonaalsed** , st skalaarkorrutis <a|b> on erinev nii ühest kui ka nullist. Teeme hüpoteesi, et eksisteerib kloonimismasin, mis algse sõltumatutest osadest koosneva liitsüsteemi otsekorrutisliku oleku muundab (läbi põimolekute) otsekorrutislikuks (st jälle sõltumatute osade) lõppolekuks vastavalt valemeile

$$|a>|toorik>|masin> \rightarrow |a>|a>|masin_a>$$
$$|b>|toorik>|masin> \rightarrow |b>|b>|masin_b> \quad , \tag{3.1}$$

kus |toorik> on Eve'i varudest võetud osakese olek, millest peale operatsiooni peab saama kloon. Indeksid *a* ja *b* eristavad masina olekut peale operatsiooni, kui kloonitav olek ja kloon on sõltumatult oma teed jätkamas. Kõik olekud on normeeritud 1-le : <toorik|toorik> =1 , <masin|masin> =1 jne.

Operatsioon peab olema unitaarne, st säilitab skalaarkorrutise. Operatsiooni sisendolekute skalaarkorrutis on

$$<masin|<toorik|<a|b>|toorik>|masin> = <a|b> \tag{3.1a}$$

ja väljundolekute skalaarkorrutis on

$$<masin_a|<a|<a|b>|b>|masin_b> = <a|b><a|b><masin_a|masin_b> \tag{3.1b}$$

Seega peab kehtima võrdus

$$<a|b> = <a|b><a|b><masin_a|masin_b> \quad , \qquad (3.2)$$

mis on võimalik ainult siis, kui

kas $<a|b> = 0$ (olekuvektorid ortogonaalsed, mis vastuolus algse eeldusega)

või $<a|b> = 1$ (olekuvektorid langevad kokku, mis vastuolus algse eeldusega, pealegi ei saa eristamatute olekutega edastada infobiti kahte erinevat väärtust "1" ja "0")

Seega on mitteklooonitavuse teoreem tõestatud.

Täiendatud - R nov 13 17:38:49 2009

4. Kvantkrüptograafia polariseeritud üksikfootonitega
Quantum Key Distribution with Single Polarised Photons

Meetodit demonstreerisid 1992.a. esmakordselt Bennet ja Brassard, kus rohelise valguse footonid moodustasid 40 cm pikkuse kvant-sidekanali vabas õhus. 1993.a. implementeeriti Genfi Ülikoolis see üks lihtsamaid kvantkrüptograafilisi protokolle optilisel fiibril ja saavutati võtme turvaline edastus juba 1km kaugusele.
Sideliin Alice ja Bob'i vahel koosneb kvantkanalist, mida mööda Alice saadab kindla väikese intervalliga laseriimpulssidest (ühekvantilisteks mahanõrgendamise vms teel) saadud üksikuid valitud polarisatsiooniga footoneid.
Meetodi idee on – lisaks üksikute kvantosakeste eelmainitud iseärasuste toetumisele – sideseansi käigus juhuslikult muuta footonite polarisatsiooni. Kui näiteks Alice kodeeriks oma teate bitijada "1"-d alati footoni horisontaalseks polarisatsiooniks (H) ning "0"-d vertikaalseks (V), ning Bob kasutaks talle saadetud footonite polarisatsiooni mõõtmiseks oma analüsaatoril H/V-orientatsiooni lugemaks maha need "1"-d ja "0", siis poleks Eve'l mingi probleem need vahelt kinni püüda, samal moel maha lugeda ja samasugused samas järjestuses tekitada ja Bob'le edasi saata. Seega pealtkuulamise takistamiseks tuleb kasutada vaheldumisi **mitteortogonaalseid** footoni polarisatsiooniseisundeid, mida Eve ei saaks kloonida. Selleks on Alice'l footonite väljundis kaks Pockels'i rakku, millega ta saab anda footoneile 4 erinevat polarisatsiooni vertikaaltelje suhtes:

kas nn püstibaasis (tähistame "+") polariseerimisel kas 0^0 st vertikaalne (tähis "|")
või 90^0 st horisontaalne (tähis "–")

või nn kaldbaasis (tähistame "x") polariseerimisel kas 45^0 st vasakule kaldu ("\")
või -45^0 st paremale kaldu ("/")

Kuna Alice ja Bob tahavad salaja luua võtit mingi klassikalise meetodiga šifreerimisele mineva "päris" teate hilisemaks edastamiseks, siis on kõige parem, kui sellise võtme saamiseks Alice tekitab täiesti juhusliku bitijada, st valib juhuslikult saadetava footoni polarisatsiooni nimetatud neljast. Tehniliselt on see teostatav, lastes iga laseriimpulsi hetkel Pockelsi rakkudele peale pingeimpulsside kombinatsiooni, mida kontrollib juhuslike arvude generaator. Parim selline oleks nt. kaks üksikfootonite abi-allikat, millest väljuvate footonite polarisatsioone mõõdetaks kaldu baasis, mispuhul saadakse absoluutselt juhuslikud ja ühesuguste tõenäosustega mõõtmistulemused. Meie kasutame allpool oma simulaatoris *MC* pseudojuhuslike arvude tekitaja-funktsiooni *rnd*( ) juhuslike 2-valikute tegemiseks (vt *JuV*( )paremal äärisel).

Seega siis sideliini protokoll oleks:

1. Alice valib iga saadetava footoni jaoks juhuslikult polarisatsioonibaasi: kas "+" või "x".

$$Abaas_k := JuV("+","x") \qquad (4.1.1)$$

Indeks *k* loendab saadetavaid footoneid, simulaatori algseisus tuleb neid 1 korraga ja seekord on $Abaas_1 = "+"$

Pane kursor valemisse (4.1.1.) ja, vajutades F9, tulista välja uus footon, vaata kuidas iga kord käitub $Abaas_1$!

2. Alice valib iga saadetava footoni kanda juhuslikult valitud bitiväärtuse kas "0" või "1".

$$Abitt_k := JuV("0","1") \qquad (4.1.2)$$

Seekord: $Abitt_1 = "1"$

Pane kursor valemisse (4.1.2.) ja, vajutades F9, tulista välja uus footon, vaata kuidas iga kord käitub $Abitts_1$!

3. Alice annab igale saadetavale footonile ühe neljast polarisatsioonist, vastavalt 2-e eelmise sammu tulemustele

$$Apol_k := if\left(Abaas_k = "+", if\left(Abitt_k = "0", "—", "|"\right), if\left(Abitt_k = "0", "\", "/"\right)\right) \qquad (4.1.3)$$

Seekord : $Apol_1 = "|"$

Uuri, kas//kuidas on $Apol_1$ vastavuses $Abaas_1$ ja $Abitt_1$ praeguste väärtustega valemi (4.1.3) kohaselt!
Pane kursor valemisse (4.1.3.) ja, vajutades F9, vaata kuidas iga kord käitub $Apol_1$! Miks $Apol_1$ ei muutu ?

4. Bob valib iga saabuva footoni jaoks juhuslikult ja Alice'st sõltumatult polarisatsioonibaasi: kas "+" või "x".

$$Bbaas_k := JuV("+","x") \qquad (4.1.4)$$

Seekord : $Bbaas_1 = "x"$

5. Bob teostab valitud baasis polarisatsiooni mõõtmise.
Kui Bob'i baas on juhtumisi sama, mis Alice'l selle footoni saatmisel, siis on Bob'i lugem identne Alice saadetud bitiga. Kui aga baasid erinevad, siis, kuna nurk nende vahel on 45 kraadi, läheb footon Bobi analüsaatoris võrdse tõenäosusega kas detektorile-fotokordistile tähisega "0" või detektorile-fotokordistile tähisega "1"" ja seda muidugi kvantmehaanilise absoluutse juhuslikkusega. Seega Bob saab tulemuse:

$$Blugem_k := if\left(Bbaas_k = Abaas_k, str2num\left(Abitt_k\right), JuV(0,1)\right)$$

Seekord : $Blugem_1 = 1$

Uuri, kas/kuidas on $Blugem_1$ vastavuses tema valemisse minevate suuruste väärtustega.
Pane kursor valemisse ja, vajutades F9, vaata kuidas iga kord käitub $Blugem_1$!
Pane kursor valemisse (4.1.4) ja, vajutades F9, vaata kuidas iga kord käitub $Blugem_1$! Millest erinevus?

6. Bob ja Alice suhtlevad läbi avaliku kanali ja teevad kindlaks, millistel juhtudel neil olid samad baasid, kuid muidugimõista Bob jätab oma mõõtmistulemused enda teada.

$$OnEkviBd_k := if\left(Bbaas_k = Abaas_k, "jah", "ei"\right)$$

Seekord : $OnEkviBd_1 = "ei"$

7. Salajane võti moodustatakse vaid nendest bittidest, mida Bob luges juhtudel, kui tema baas oli sama, mis Alice'l vastavate footonite saatmisel, sest neil juhtudel on Alice ja Bob'i bitiväärtused samad. Ülejäänud footoneid ignoreeritakse võtme moodustamisel, aga kasutatakse kvantkanali testimiseks.

$$V6ti_k := if\left(OnEkviBd_k = "jah", num2str\left(Blugem_k\right), " "\right)$$

Seekord : $V6ti_1 = " "$

Meie simulaatoris juhul kui baasid erinevad, tuleb *V6ti* väärtuseks tühik, st võtmesse kui 1-de ja 0-ide jadasse see footon kahendkohta juurde ei andnud.

Pane kursor määratlusse *FootoniteArv* ja, vajutades F9, lase Alice'l saata järjekordne footon!
Uuri eelnevaid andmeid kokkuvõtvat tabelit allpool ja kontrolli, kas võtmesse sobivatel juhtudel on Bob mõõtnud maha sama numbri, mida Alice saatis. Siis tekita footonite seeria *FootoniteArvuga* 15 ja korda uurimist. Võib ka pikemaid seeriaid kasutada, siis tuleb tabelit *scrolli* 'da.

FootoniteArv ≡ 1

Tbl =

"Abaas"	"Abitt"	"Apol"	"Bbaas"	"Blugem"	"OnEkviBd"
"+"	"0"	"—"	"+"	0	...

Siiani pole me arvestanud Eve võimalikku sekkumist. Kuna Eve ei tea, missugust baasi Alice kasutas, saab ta valida vaid juhusliku baasi oma mõõtjal, mis on näiteks samasugune kui Bob'i oma. Ta ei saa ka teada, millal tema baas oli vale, sest ühefootonilise pulsi mõõtmisel reageerib vaid üks kahest fotodetektorist st saadakse vaid üks lugem. Näiteks Eve aparatuuri "1"-kanali detektori "kliks" võib tähendada nii seda, et tema mõõtebaas oli sama, mis Alice'l ja seega Alice saatis tõesti "1"-e, kuid võib tähendada ka seda, et baasid olid üksteise suhtes kaldu ja footon "otsustas" juhtumisi Eve aparaadi kaltsiidikristallis minna detekteerimiskanalisse "1" . Seega on Evel valida vaid mitmesuguste juhusliku tegutsemise strateegiate vahel, mis kõik aga on Alice ja Bob'i poolt avastatavad. Näiteks võib Eve saata Bobile edasi lihtsalt oma aparaadi kaltsiidikristallist väljuva footoni, tehes fotokordistis neeldumise asemel peenemate meetoditega kindlaks, **kumba** kanalit footon läbis. Siiski on see edasisaadetud footon mõõtmise tulemusel oma olekut muutnud ja tekitab seega Alice ja Bob'i tulemuste hulgas vigu, sealhulgas neis, mis võtme moodustamisest nagunii üle jäid. Vastava statistilise töötlemisega kvantmehaanika valemite toel saavad Alice ja Bob mitte ainult teada Eve tegevusest, vaid ka rakendada strateegiat, mis suvaliselt suure soovitava tõenäosusega tagab võtmeedastuse turvalisuse.

Siin me vastava protokolli detailidesse ei lasku, Eve avastamist käsitleme lähemalt krüptograafiaprotokollis, mis kasutab põimolekus footonite paare (vt põimolekus kvantbitt ingl. k. *entangled qubit* ehk *ebit* **esimeses töölehes**).

Küsimused

1. Miks saab valemeis (3.1a) ja (3.1b) ket-vektorite järjestust omavahel ja bra-vektorite järjestust omavahel muuta, nagu neis valemeis on tehtud.
2. Miks mitteklоonitavuse teoreemis tegeldakse mitte-ortogonaalsete seisunditega?
3. Miks salajase võtme-bitijada moodustamiseks A ja B kasutavad vaid neid tulemusi, kus nende mõõtmisbaasid olid identsed?
4. Mismoodi nad teevad kindlaks, milliste footonite jaoks need baasid olid identsed, mis tingimusi peaks täitma vastav sidekanal?
5. Milles võiks seisneda mittevõltsitava "kvantraha" idee?
6. Kui pikk on tänapäeval realiseeritud (üksikfootonitega) valguskiud-sideliini pikkus ja mis on siin põhimõtteliseks piiranguks (otsi materjali Internetist).

------- töölehe esimese poole lõpp ----------

Kvantkrüptograafiast (järg)

Sisukord

Täiendatud - P jaan 17 16:39:27 2010

5. Kvantkrüptograafia põimseisundis footonipaaridega
Quantum Key Distribution with Entangled States

Üksikfootonitega kvantkrüptograafia (või kitsamalt – kvant-võtmekehtestus) võib polarisatsiooni asemel kasutada ka näiteks info kodeerimist footonite faasi. Veelgi laiem valik võimalusi tekib osakesepaaride kasutamisel, sest põimolekuis võivad osakesed olla väga mitmesuguste füüsikaliste suuruste/vabadusastmete järgi. Järjepidevuse ja võrdlusvõimaluse huvides vaatleme siin põimolekuid polarisatsiooni järgi.

Olgu meil footoni polarisatsiooni omaolekud polarisatsioonimõõtmise baasis "+" (Hor. / Vert.)

$$|H\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |V\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad .$$

Siis footonipaari kui kahe kvantbiti süsteemi omaolekud on otsekorrutised (esimese töölehe eeskujul):

$$|HH\rangle := |H\rangle \otimes |H\rangle \qquad |HV\rangle := |H\rangle \otimes |V\rangle \qquad |VH\rangle := |V\rangle \otimes |H\rangle \qquad |VV\rangle := |V\rangle \otimes |V\rangle$$

$$|HH\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad |HV\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \qquad |VH\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \qquad |VV\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Alice ja Bob saavad kumbki ühe footoni paarist, mida kirjeldab vertikaalselt ja horisontaalselt polariseeritud footonite põimolek kujul:

$$|\ \rangle := \frac{1}{\sqrt{2}}(|HV\rangle - |VH\rangle) \qquad |\ \rangle = \begin{pmatrix} 0 \\ 0.7071 \\ -0.7071 \\ 0 \end{pmatrix}$$

Selgituseks: $\frac{1}{\sqrt{2}} = 0.7071$

Ütleme, et Alice saab vasakpoolse ja Bob parempoolse. Kui nüüd Alice mõõdab oma footoni polarisatsiooni, kollapseerub liitsüsteemi olek, nagu ikka, ühte oma omaolekusse, st antud juhul vaid kas olekusse $|HV\rangle$ või $|VH\rangle$, sest ülejäänud kahe omaoleku (neljast) amplituud superpositsioonis on null.

See aga tähendab jällegi meile juba Shor'i algoritmist tuntud põimoleku väljaprojekteerumist, st kui Alice footon detekteerub horisontaalse polarisatsiooni kanalis, siis samal hetkel, sõltumata distantsist, Bob'i footonil ei jää muud üle, kui "olla vertikaalne". Ja vastupidi, kui Alice'l juhtumisi "V", siis Bob'l kindlalt "H".

Nagu eelmiseski protokollis, kui Alice ja Bob piirduksid "+"-baasi kasutamisega, oleks Eve töö lihtne. Seepärast jällegi Alice ja Bob valivad iga footoni jaoks juhuslikult erineva mõõtebaasi, mis kaldu mingi nurga all vertikaaltelje suhtes.

Siinkohal on aga seetõttu meil vaja väljendada vaadeldud põimolek teises baasis – nurga α all kaldu baasis, sest Alice või Bob'i polarisatsioonimõõturite omaolekud pole enam |H> ja |V> , vaid nende pööratud versioonid. Pöördeteisenduse maatriks, nagu teada, avaldub

$$U(\alpha) := \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ \sin(\alpha) & -\cos(\alpha) \end{pmatrix}$$

, kusjuures, nagu ikka koordinaatide teisendamise puhul, on tulemus sama, kui pöörata telgi α võrra või hoopis objekti-vektorit $-\alpha$ võrra.

Seega defineerime uued pööratud baasivektorid/omaolekud, mistõttu algse "+"-baasi omavektorid |H> ja |V> pole enam komponetidega 1 ja 0 , vaid midagi vahepealset, sõltuvalt pöördenurgast α :

$$|H\rangle(\alpha) := U(\alpha)\cdot|H\rangle \qquad |H\rangle(\alpha) \rightarrow \begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \end{pmatrix}$$

$$|V\rangle(\alpha) := U(\alpha)\cdot|V\rangle \qquad |V\rangle(\alpha) \rightarrow \begin{pmatrix} \sin(\alpha) \\ -\cos(\alpha) \end{pmatrix}$$

Nüüd saame väljendada algse põimoleku Alice ja Bob'i eri nurkade α ja β alla pööratud mõõturite omaolekute kaudu, kusjuures põimoleku selline esitus jääb sõltuma nurkadest kui parameetritest:

$$|\ \rangle(\alpha,\beta) := \frac{1}{\sqrt{2}}\cdot(|H\rangle(\alpha)\otimes|V\rangle(\beta) - |V\rangle(\alpha)\otimes|H\rangle(\beta))^{\blacksquare} = \begin{bmatrix} \frac{1}{2}\cdot(-\sin(\alpha)\cdot\cos(\beta) + \cos(\alpha)\cdot\sin(\beta))\cdot 2^{\frac{1}{2}} \\ \frac{1}{2}\cdot(-\cos(\alpha)\cdot\cos(\beta) - \sin(\alpha)\cdot\sin(\beta))\cdot 2^{\frac{1}{2}} \\ \frac{1}{2}\cdot(\sin(\alpha)\cdot\sin(\beta) + \cos(\alpha)\cdot\cos(\beta))\cdot 2^{\frac{1}{2}} \\ \frac{1}{2}\cdot(-\sin(\alpha)\cdot\cos(\beta) + \cos(\alpha)\cdot\sin(\beta))\cdot 2^{\frac{1}{2}} \end{bmatrix}^{\blacksquare}$$

...saab Alice ja Bobi baasis (mis määratud kumbagi poolt kasutatava mõõteriista orientatsiooninurgaga α ja β , vastavalt) väljendada sama põimoleku lihtsamalt:

$$|\ \rangle(\alpha,\beta) := -\frac{1}{\sqrt{2}}\cdot\begin{pmatrix} \sin(\alpha-\beta) \\ \cos(\alpha-\beta) \\ -\cos(\alpha-\beta) \\ \sin(\alpha-\beta) \end{pmatrix} . \qquad (5.1)$$

Näeme, et suvaliste nurkade puhul on põimoleku vektoril Alice&Bob'i baasis 0-st erinevad kõik 4 amplituudi/komponenti, st – superpositsioonis on esindatud kõik baasivektorid, mitte ainult kaks.

Seejuures aga juhul, kui Alice ja Bob teevad oma mõõtmisi ühtemoodi kaldu baasis (st $\alpha = \beta$), on nende jaoks tulemus sama, nagu olnuks tegu algse põimolekuga ja selle mõõtmisega "+"-baasis. Teiste sõnadega, kui $\alpha = \beta$, siis kaldest sõltumata on Alice ja Bob'i mõõtmistulemused antikorreleeritud: kui A saab tulemuse "H", siis B saab kindlalt tulemuse "V" ja vastupidi.

Edasise analüüsi mugavuse huvides seostame tulemusega "H" omaväärtuse +1 ja tulemusega "V" – -1. Võime ka öelda, et mõõdetakse füüsikalist suurust "polarisatsiooni rõhtsus", millel vaid kaks väärtust (vertikaalselt polariseeritud footon oleks siis nagu negatiivse rõhtsusega).

Üldjuhul, kui Alice ja Bob teevad oma mõõtmisi erimoodi kaldu baasides (st $0 \neq \alpha \neq \beta \neq 0$), pole mõõtmistulemused täiesti antikorreleeritud. Millisel määral nad seda on, saab kirjeldada Alice ja Bob'i mõõdetud "rõhtsuste" korrutise keskväärtusega. Kui mõlemad saavad tulemuse "H" , siis see korrutis on (+1)(+1) =1 , kui tulemused "H" ja "V", siis see korrutis on (+1)(-1) = -1, jne. Iga korrutise väärtuse saamise tõenäosus on – vastavalt kvantmehaanilise olekuvektori komponentide tähendusele – võrdne komponendi ehk amplituudi mooduli ruuduga (meie juhul lihtsalt ruuduga, sest meil siin pole mängus kompleksarvulisi amplituude). Seega, keskväärtuse üldise definitsiooni kohaselt, footonipaari "rõhtsuste" korrutise keskväärtuse leidmiseks tuleb summeerida kõik 4 võimalikku väärtust, igaüks võetud kaaluga, mis võrdne just selle väärtuse esinemise tõenäosusega. Selline summa avaldub ja lihtsustub alljärgnevalt.

$$\left(|\ \rangle(\alpha,\beta)_1\right)^2 - \left(|\ \rangle(\alpha,\beta)_2\right)^2 - \left(|\ \rangle(\alpha,\beta)_3\right)^2 + \left(|\ \rangle(\alpha,\beta)_4\right)^2 \text{ simplify } \rightarrow 2\cdot\sin(\beta-\alpha)^2 - 1$$

Kuna trigonomeetria teisendusvalemeist $-\cos(2\gamma)\ \text{expand}\ \rightarrow -\cos(2\cdot\gamma)$,

siis korrelatsioonikoefitsient Alice'i ja Bobi mõõtmistulemuste ±1 vahel avaldub lihtsal kujul

$$\text{KorrlnAB}(\alpha,\beta) := -\cos[2(\alpha-\beta)] \quad . \tag{5.2}$$

Nüüd tuleme konkreetsete $22{,}5^0$ sammuga nurkadekombinatsioonide juurde, mis sobivad nii kvantmehaanika lõpliku õigsuse *experimentum crucis*'ele kui ka kvant-võtmekehtestuse protokollile

$$\alpha_1 := 0 \qquad \alpha_2 := \frac{1}{4}\cdot\pi \qquad \alpha_3 := \frac{1}{8}\cdot\pi \qquad \alpha_1 = 0\cdot\text{deg} \quad \alpha_2 = 45\cdot\text{deg} \quad \alpha_3 = 22.5\cdot\text{deg}$$

(5.3)

$$\beta_1 := 0 \qquad \beta_2 := -\left(\frac{1}{8}\cdot\pi\right) \qquad \beta_3 := \frac{1}{8}\cdot\pi \quad \text{, ehk} \qquad \beta_1 = 0\cdot\text{deg} \quad \beta_2 = -22.5\cdot\text{deg} \quad \beta_3 = 22.5\cdot\text{deg}$$

Moodustame (5.2)-tüüpi suurustest avaldise

$$\text{BellKritrm} := \text{KorrlnAB}\left(\alpha_1,\beta_3\right) + \text{KorrlnAB}\left(\alpha_1,\beta_2\right) + \text{KorrlnAB}\left(\alpha_2,\beta_3\right) - \text{KorrlnAB}\left(\alpha_2,\beta_2\right) \tag{5.4}$$

On tõestatud, et eri nurkadega mõõtmiste tulemustest niimoodi moodustatud korrelatsioonide-avaldise (5.4) väärtus peab jääma

vahemikku $-\sqrt{2} \le \text{BellKritrm} \le \sqrt{2}$, kui kvantmehaanika iseenesest õigete ennustuste mõtestamises vaja tugineda nn varjatud parameetritele vms, mis nagu juhivad kvantmehaanilist juhuslikkust

ja

peab olema võrdne

$\text{BellKritrm} = -2\cdot\sqrt{2}$, kui kvantmehaanika on õige nii nagu ta on.

Kvantkrüptograafia terminites tähendab see seda, et kui avaldis (5.4) on sideseansi käigus statistiliselt oluliselt erinev (vähem negatiivne) väärtusest $\text{BellKritrm} = -2\cdot\sqrt{2}$, siis on Eve kui klassikaline subjekt tegutsemas kvantkanali kallal....

Nüüd on meil piisav teoreetiline põhi all asumaks tutvuma võtmekehtestusprotokolli endaga (simulaatori abil).

Protokoll lähtub sellest, et teadaolevatel ajahetkedel (teadaoleva taktsagedusega) saavad Alice ja Bob kumbki ühe footoni põimolekus $\frac{1}{\sqrt{2}}(|HV\rangle - |VH\rangle)$ footonipaarist. Nende allikas võib asuda näiteks Alice juures (mispuhul Bob saab oma footoni kvantkanali pikkusest sõltuva viivisega). Nende kasutada on ka klassikaline sidekanal moonutustekindlaks infovahetuseks kasutatud polarisatsioonimõõtebaaside (nurkade) kohta, mida nad valivad juhuslikult 3-st määratud väärtusest. Kokkulangevate baaside puhul saadud mõõtmistulemustest moodustub salavõti, kusjuures "õudse kaugmõju" tõttu Alice ja Bob teavad, et ükskõik kumb neist esimesena mõõtis ja mingi tulemuse sai, siis teine saab kindlalt alternatiivse tulemuse . Ülejäänud footonipaaridel saadud tulemusi kasutatakse testimaks, kas Eve pole kvantkanali kallal tegutsemas.

Mõõtebaasi määramiseks on Alice'l ja Bob'il vaja sõltumatult 3-st võimalusest täiesti juhusliku valiku tekitajat tõenäosustega 1/3, 1/3, 1/3 iga valiku jaoks. Meie kasutame selleks pseudojuhuslike arvude generaatorfunktsiooni *rnd*( )

Seega protokoll oleks järgmine.

1. Nii Alice kui ka Bob valivad iga tuleva footoni jaoks juhuslikult oma polarisatsioonibaasi – vastavalt ühe kaldenurga kolmest kumbalegi etteantust

$Anurk_1 := Ju3V(\alpha_1, \alpha_2, \alpha_3)$ $Bnurk_1 := Ju3V(\beta_1, \beta_2, \beta_3)$

Indeks m loendab saadavaid footonipaare, simulaatori algseisus tuleb neid 1 korraga.

Pane kursor valemeisse ja, vajutades F9, tulista välja uus footonipaar, vaata nurkade väärtusi! Seekord need on:

$Anurk_1 = 0$ $Bnurk_1 = 0$ ehk kraadides $Anurk_1 = 0 \cdot deg$ $Bnurk_1 = 0 \cdot deg$

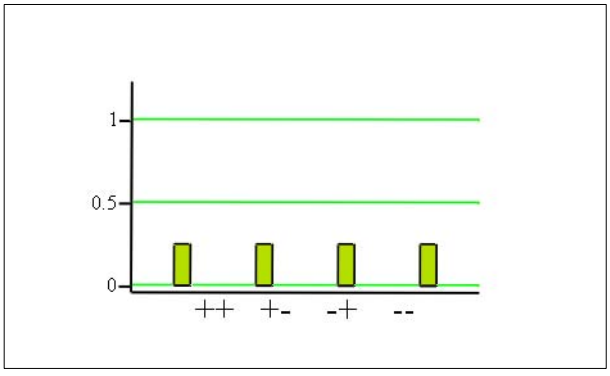
2. Nii Alice kui ka Bob mõõdavad enda valitud baasis enda footoni polarisatsiooni.

Kokku on nende mõõtmistulemuste kombinatsioone 4, igal oma tõenäosus, mis määratud footonipaari põimoleku teisendatud vektori (5.11) komponentidega. Tõenäosuste jaotustabel koosneb 4-st arvust, mis on kvantmehaanika reeglite kohaselt vektori (5.11) nelja komponendi mooduli ruudud. Leiame jaotustabeli *m*-inda footonpaari (olgu siin *m* = 2) kohta, mille jaoks kaldenurgad olgu:

$Anurk_m := Ju3V(\alpha_1, \alpha_2, \alpha_3)$ $Bnurk_m := Ju3V(\beta_1, \beta_2, \beta_3)$

ehk seekord $Anurk_2 = 45 \cdot deg$ $Bnurk_2 = 0 \cdot deg$ kusjuures $NurkAjaBbaasiVahel_2 = 45 \cdot deg$

Vektori (5.11) mooduli ruutudest jaotustabeli $Tn4jaotus^{\langle 2 \rangle} := \overrightarrow{\left(\left|\,|\rangle(Anurk_2, Bnurk_2)\right|\right)^2}$ ja seda kujutava tulpdigrammi arvutamiseks kasutame *MC* maatriksoperatsioone



Diagrm

Pane kursor ühele kollastest avaldistest ülalpool, vajuta F9 ja vaata kuidas koos nurkadega muutub Alice'i ja Bob'i tulemusvariantide ++, –+, +– ja – – tõenäosus.

Edaspidi saab meil olema suvaline arv *m* footonpaare, see arv võib olla ka väga suur. Kordame tõenäosuste arvutamise valemit-definitsiooni suvalise *m* jaoks

$$\mathrm{Tn4jaotus}^{\langle m\rangle} := \overrightarrow{\left(\left|\ |\ >\left(\mathrm{Anurk}_m, \mathrm{Bnurk}_m\right)\right|\right)^2}$$

Meil on vaja aga ka mõõtmisaktis toimuva (kvantmehaanilise juhuslikkusega) ühte omaolekusse kollapseerumise/kokkukukkumise simulaatorit, mis valiks ühe omaoleku/mõõtmistulemuse neljast vastavalt tõenäosuste jaotusele.

$$\mathrm{yks4st}_m := \mathrm{OmaolekuNr\_kuhukukubOlekmillel}\left(\mathrm{Tn4jaotus}^{\langle m\rangle}\right)$$

Pane kursor *yks4st*-avaldisele, vajuta F9, kordamaks A&B mõõtmisi samade baaside korral.

Seekord on valik selline $\mathrm{yks4st}_2 = 2$

Üks kahest esimesest omaolekust (++ või +–) tähendab Alice mõõtmistulemust +1, seepärast

$\mathrm{Atulem}_m := \mathrm{if}\left(\mathrm{simuleerime} = \text{"kvant.meh."}, \mathrm{if}\left(\mathrm{yks4st}_m \le 2, 1, -1\right), \mathrm{JuV}(1,-1)\right)$. Seekord $\mathrm{Atulem}_2 = 1$

Bob'i mõõtmistulemus +1 tuleb aga siis, kui kollapseerumine toimub paaritu numbriga omaolekuse neljast. Seepärast simuleerime Bobi mõõtmistulemuse saamist nii:

$$\mathrm{Btulem}_m := \mathrm{if}\left(\mathrm{simuleerime} = \text{"kvant.meh."}, \mathrm{if}\left(\mathrm{trunc}\left(\frac{\mathrm{yks4st}_m}{2}\right) \ne \frac{\mathrm{yks4st}_m}{2}, 1, -1\right), \mathrm{JuV}(1,-1)\right)$$

Seekord $\mathrm{Btulem}_2 = -1$

3. Alice ja Bob teevad läbi avaliku kanali suheldes kindlaks, millised olid need footonipaarid, mispuhul nende mõõtebaasid olid identsed, st kaldenurgad võrdsed. Need mõõtmistulemused rühmitatakse salajastena, mis võetakse arvesse bitiväärtustena salajase võtme moodustamiseks. Seega iga footonipaari m kohta otsustatakse, võtta ta võtme moodustamisesk või ei:

$\mathrm{V6tta}_m := \mathrm{if}\left(\mathrm{Bnurk}_m = \mathrm{Anurk}_m, \text{"jah"}, \text{"ei"}\right)$. Seekord: $\mathrm{V6tta}_2 = \text{"ei"}$

Selles tulemusterühmas, mis võetakse võtmeks, peab kehtima antikorrelatsioon, sest eksivalentsete baaside korral, nagu ülapool veendusime, mõõdavad Alice ja Bob (ka nende baasis vaid kahe nullist erineva komponendiga 4st) maksimaalselt põimitud olekut $\frac{1}{\sqrt{2}}(|HV> - |VH>)$

Kontrollime, kas tulemused ikka on vastasmärgilised, st antikorreleeruvad

$\mathrm{OnAeiv6rduB}_m := \mathrm{if}\left(\mathrm{Btulem}_m \ne \mathrm{Atulem}_m, \text{"jah"}, \text{"ei"}\right)$

Seekord $\mathrm{OnAeiv6rduB}_2 = \text{"jah"}$

Salajase võtme bitiväärtusteks lähevad kirja Alice numbrid (või Bob'i omad "–"märgiga).

$$\mathrm{V6tmeks}_m := \mathrm{if}\left(\mathrm{V6tta}_m = \text{"jah"}, \mathrm{num2str}\left(\frac{\mathrm{Atulem}_m + 1}{2}\right), \text{" "}\right)$$

Seekord $\mathrm{V6tmeks}_2 = \text{" "}$

Protokolli jaoks olulised andmed on kokku võetud siin all tabelis. Tõstes footonipaaride arvu 15-ni või enam, vajuta F9, uuri ja kõrvuta erinevate paaride peal saadud andmeid.

FootonipaarideArv ≡ 1

	1	2	3	4	5	6	7
1	"Abaas [rad]"	"Bbaas [rad] "	"A tulem"	"B tulem"	"AxB = -1?"	"Võtta?"	"Võtmeks"
2	0.785	0	1	-1	"jah"	"ei"	" "
3							
4							

Tbl =

4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

Tabeli kerimiseks allapoole, kui *m* >16, kõpsa tabelil suvalises kohas.

Edasi uurime, kuidas avastada Eve'i tegevust. Eve sekkumise lihtsamaid strateegiaid taandub sellele, et Alice'i ja Bob'i mõõtmistes kvantmehaanikale vastav juhuslikkus asendub n.-ö. varjatud parameetrile iseloomuliku klassikalise juhuslikkusega (nagu oleme sisse kirjutanud *Atulem* ja *Btulem* määratlustesse).

simuleerime ≡ "kvant.meh."

Anna muutujale *simuleerime* väärtuseks "Eve'i tegevust" või "Varjat.param.-ga teooriat" ja uuri muutusi Tabeli sisu seaduspärasustes.
Seejärel **taasta** siit lause lõpust kopeerimise teel muutuja *simuleerime* väärtus "kvant.meh." !
Pane footonipaaride arvuks 300 kuni 10000 või enam, olenevalt oma arvuti võimsusest (alusta tagasihoidlikult veendumaks, et Sinu arvuti Tabeli välja arvutaks mitte kauem kui paarikümne sekundiga).

4. Alice ja Bob informeerivad läbi avaliku kanali üksteist võtme moodustamisest üle jäänud mõõtmistulemustest testimaks kanali turvalisust. Osa kõikvõimalikest erinevate baasidega saadud mõõtmistulemustest võimaldavad arvutada toimunud mõõtmisteseeria jaoks n.-ö. eksperimentaalse Belli kriteeriumi (vt valem 5.4). Piisavalt suurearvulise seeria puhul peab arvutustulemus olema lähedane Belli kriteeriumi teoreetilisele väärtusele $-2\cdot\sqrt{2}$. Kui ta aga (statistiliselt usaldusväärselt) on erinev (positiivsem teoreetilisest), siis on Eve sekkunud ja tuleb võtta lisameetmeid turvalisuse tagamiseks.

Teeme testi läbi simulaatoril (kui oled taastanud simulaatori kvantmehaanilise režiimi eelmises lõigus)

Võtmeiks võetuist üle jäänud footonipaaride arv on

$$\text{TestiksFtpaare} := \sum_{m=2}^{\text{length}(V6tta)} \left(V6tta_m = \text{"ei"}\right)$$

Seekord

$$\text{length}(V6tta) - 1 = 1$$

$$\text{TestiksFtpaare} = 1$$

Neist on Belli kriteeriumi valemi (5.4) kohaselt vaja võtta vaid 4-le nurkadekombinatsioonile vastavad:

$Anurk1Bnurk3Arv := \sum ON1ja3$ Seekord

$Anurk1Bnurk3Arv = 0$

$Anurk1Bnurk2Arv := \sum ON1ja2$ Seekord

$Anurk1Bnurk2Arv = 0$

$Anurk2Bnurk3Arv := \sum ON2ja3$ Seekord

$Anurk2Bnurk3Arv = 0$

$Anurk2Bnurk2Arv := \sum ON2ja2$ Seekord

$Anurk2Bnurk2Arv = 0$

$BellKritrm \to -2\cdot\sqrt{2}$

Alice'i ja/või Bob'i poolt vastavalt valemile (5.4) moodustatud andmetöötlusavaldisele (vt paremäärisel ☞) välja arvutatud väärtus on seekord

$BellKritrm_{experim} = 0$

, mida tuleb võrrelda "õige" väärtusega ☞ (muidugi pole võrdlusel mõtet statistika jaoks väikeste footonipaaride arvu puhul)

$-2\cdot\sqrt{2} = -2.8284$.

$FootonipaarideArv = 1$

Tekita F9-ga kursori olles *FootonipaarideArv*'u määratlusel uus võtmekehtestusseanss ja hinda testitulemusi – kas head või kahtlased?

Siis anna muutujale *simuleerime* väärtuseks "Eve'i tegevust" või "Varjat.param.-ga teooriat" ja vaata, kuhukanti nüüd *BellKritrm*'i väärtus kukub! Tee oma järeldused!

6. Kvant-tihekodeerimise protokoll
Quantum Dense Coding Protocol

Eelmine protokoll kasutas osakesepaari kui kvantbitipaari ühte maksimaalselt põimitud seisundit

$|\ > := \frac{1}{\sqrt{2}}(|HV> - |VH>)$, milliseid aga on veel – kokku 4 nn Belli põimseisundit:

1. $|s+> := \frac{1}{\sqrt{2}}(|HV> + |VH>)$

2. $|s-\ > := \frac{1}{\sqrt{2}}(|HV> - |VH>)$

3. $|t+> := \frac{1}{\sqrt{2}}(|HH> + |VV>)$

4. $|t-> := \frac{1}{\sqrt{2}}(|HH> - |VV>)$

Mitmendat Belli seisundit kasutasime eelmises punktis?

Võrdluseks – 2 asemel kõigist neljast baasivektorist võrdse amplituudiga kokku pandud olek

$$\frac{1}{2}\cdot(|HH\rangle + |HV\rangle + |VH\rangle + |VV\rangle) = \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{pmatrix}$$

ei ole põimseisund,
sest esitub 45^0 kaldolekute otsekorrutisena:

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \otimes \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{pmatrix}$$

Meenuta esimest töölehte – mitte iga **2-st** baasivektorist võrdse amplituudiga kokku pandud olek pole põimseisund. Too näide!

Belli põimseisundid on kasutatavad väga mitmetes kvant-informaatika-alastes protokollides, algoritmides, jt rakendustes (nt. osakeste seisundi kvant-teleportatsioon). Vaatleme selles punktis lühidalt kvant-(super)tihekodeerimist (*quantum superdense coding*), mis nagu kvant-krüptograafiagi, kuulub kvant-telekommunikatsiooni valdkonda.

Klassikalises sideliini puhul on 4 biti info edastamiseks vaja saata 2 bitikandjat, näiteks 2 elektriimpulssi, millel kumbalgi võimalikud 2 pingenivood: +1 ja -1.
Osutub, et kasutades kodeerimiseks nelja Belli seisundit, on võimalik kvantsideliinis edastada **4 bitti vaid ühe kvantbitiga-footoniga**, st tihendusfaktoriga 2 korda võrreldes klassikalise piiriga.
Vastav protokoll on järgmine.

1. Alice ja Bob saavad kumbki ühe footoni põimfootonite paarist, mis 1.-s Belli olekus $|s+\rangle$
   Alice hoiab endale tulnud footonit (viiteliinis vms. nt. nanosekundi) kuni Bob teostab kodeerimise.

2. Bob kodeerib oma 4-bitise teate oma footoni polarisatsiooni, mille variante olgu 4. Olekuvektorite terminites tehku Bob oma footoniga-kvantbitiga (paariolekus kirjas paremal) **kas**:
   2.1 eimidagi – unitaarne teisendus ühikmaatriksiga ja paari olek jääb endiseks – $|s+\rangle$ **või**
   2.2 ümberviske – unitaarne teisendus 1.-e Pauli maatriksiga ja paari olek $|s+\rangle \rightarrow |t+\rangle$ **või**
   2.3 faasinihke – unitaarne teisendus 3.-nda Pauli maatriksiga ja paari olek $|s+\rangle \rightarrow |s-\rangle$ **või**
   2.4 ümberviske ja faasinihke – unitaarne teisendus 2.-e Pauli maatriksiga ja paari olek $|s+\rangle \rightarrow |t-\rangle$.

3. Bob saadab oma footoni Alice'le, kes määrab, millises 4.-st Belli olekust footonipaar oli.

Seega on Alice saanud Bob'ilt ühe kvantbitiga 2 bitti informatsiooni.

------- Töölehe lõpp -------

Kvant-teleportatsioon

Teleportatsioon on teaduslikust fantastikast tuntud protseduur, millega objekt taasluuakse kauges ruumipunktis kui täpne koopia originaalist lähtepunktis, kusjuures läbi ruumi saadetakse vaid originaali kohta informatsiooni kandev signaal ning originaal ise lõpetab eksistentsi.
Kvant-teleportatsiooni objektiks on kvantsüsteemi seisund -- lihtsaimal juhul kvantbiti seisund.
Seejuures vastavalt kvantmehaanika põhitõdedele igasugune kvantsüsteemi seisundi mõõtmine saatja poolt üldjuhul hävitab selle seisundi, ilma et saadaks tema kohta kogu seda informatsiooni, mis on vajalik seisundi rekonstrueerimiseks vastuvõtja juures.
Sajandivahetuse paiku teostati kvant-teleportatsioon ka eksperimentaalselt lühemate ja pikemate vahemaade taha nii footonitega kui ka aatomitega (kui kvantbiti realisatsioonidega).

Käesolev tööleht-simulaator tutvustab kvant-teleportatsiooni ideed ja protokolli.

Alice teleporteerib Bob'ile kvantbiti, mis on suvalises superpositsioonilises seisundis. Millises nimelt, st missugused on olekuvektori 2 komponenti või 2 nurka Blochi sfääril, seda Alice ega üldse keegi ei tea.

1. Alice loob kvant-teleportatsiooni kanali

Salastatud - Thu Dec 13 22:01:10 2007

Teleporteerimisele minev tundmatu seisund on $|\psi\rangle := \begin{pmatrix} a \\ b \end{pmatrix}$. (1)

Millised on kompleksarvude *a* ja *b* väärtused, pole teada, aga eeldame seisundi normeeritust: $\langle\psi\cdot|\psi\rangle = 1$.

Lisaks teleporteeritavale kvantbitile (nr.1) on vaja 2 abi-kvantbitti (nr. 2, 3). Olgu konkreetsuse mõttes kvantbitid realiseeritud footonite polarisatsoonil horisontaalne/vertikaalne. Tähistame vastavad omaolekud -- baasivektorid:

$$|H1\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |V1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad |H2\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |V2\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad |H3\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |V3\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Alice moodustab abibittidest põimseisundi -- nn põimbiti (ingl. k. *ebit*). Kvantarvuti formaalsetes terminites laseb ta nad selleks läbi XOR-lüli (vt. 1. tööleht), füüsikaliselt aga need footonid luuaksegi põimseisundis ühe suure energiaga footoni lagunemisel kaheks spetsiaalses mittelineaarses kristallis.

$$|\,\text{ebit}\,\rangle := \frac{1}{\sqrt{2}}(|H2\rangle \otimes |V3\rangle - |V2\rangle \otimes |H3\rangle)$$

See on üks eelmistest töölehtedest tuntud põimseisundeist. Põimitus paistab välja ka tema komponentide väärtustest:

$$\sqrt{2}\cdot|\,\text{ebit}\,> = \begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix}$$

Mismoodi saab siit veenduda, et see on põimseisund?

Alice käsutuses olev teleporteeritav kvantbitt (nr. 1) pole kuidagi seotud tekitatud põimbitiga, mistõttu kolme biti seisund -- vaadelduna ühisena -- on otsekorrutisseisund biti nr. 1-ga:

$$|123> = |\psi> \otimes |\,\text{ebit}\,> = |\psi> \otimes \left[\frac{1}{\sqrt{2}}(|H2> \otimes |V3> - |V2> \otimes |H3>)\right] \quad \text{, kuna aga} \quad |\psi> = a\cdot|H1> + b\cdot|V1>\,,$$

siis

$$|123> := \frac{a}{\sqrt{2}}[|H1> \otimes (|H2> \otimes |V3>) - |H1> \otimes (|V2> \otimes |H3>)] + \frac{b}{\sqrt{2}}\cdot[|V1> \otimes (|H2> \otimes |V3>) - |V1> \otimes (|V2> \otimes |H \quad (2)$$

kus ümarsulud pole olulised, sest otsekorrutis on assotsiatiivne.

Alice saadab ühe abibittidest (näiteks nr.3-e) Bobile, kui distantsist hoolimata jääb see põimituks Alice juurde jäänd paarilisega.

2. Alice teostab enda kvantbitipaaril mõõtmise Bell'i seisundi analüsaatoriga ja edastab tulemuse Bobile läbi klassikalise sidekanali.

Kõigepealt täpsustame mõõtmise protseduuri, mida Alice peab teleporteerimise jaoks tegema. Mõõtmisele tuleb allutada korraga 2 kvantbitti -- nr.1 (teleporteeritava seisundi kandja) ja nr.2 --, st oleku kokkukukkumine toimub 4-mõõtmelises Hilberti ruumis, milles baasivektorid on

$$|HH> := |H1> \otimes |H2> \qquad |HV> := |H1> \otimes |V2> \qquad |VH> := |V1> \otimes |H2> \qquad |VV> := |V1> \otimes |V2>$$

$$|HH> = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad |HV> = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \qquad |VH> = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \qquad |VV> = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Lühiduse huvides jätsime kvantbittide nummmerduse baasivektorite tähistuses ära -- silmas on peetud siin ja edaspidi nr.1 ja nr. 2.
Nagu geomeetriastki juba teada, neist vektoreist moodustatud suvaline ortogonaalne lineaarkombinatsioon on samavõrd kasutatav baasivektorite nelikuna. Võtame selleks Belli seisundid, mis järjestatud siin natuke teisiti kui eelmisel töölehel.

$$\underset{1.}{|s->} := \frac{1}{\sqrt{2}}(|HV> - |VH>) \qquad \underset{2.}{|s+>} := \frac{1}{\sqrt{2}}(|HV> + |VH>) \qquad \underset{3.}{|t->} := \frac{1}{\sqrt{2}}(|HH> - |VV>) \qquad \underset{4.}{|t+>} := \frac{1}{\sqrt{2}}(|HH> + |VV>)\ . \quad (3)$$

Uutes lühitähistustes $|HH>$ jne saame kolmikseisundi $|123>$ panna kirja lühemalt (vt järgneva valemi1. rida).

$$|123> := \frac{a}{2\cdot\sqrt{2}}\cdot(|HH> \otimes |V3> - |HV> \otimes |H3>) + \frac{b}{2\cdot\sqrt{2}}\cdot(|VH> \otimes |V3> - |VV> \otimes |H3>) \ldots$$
$$+ \frac{a}{2\cdot\sqrt{2}}\cdot(|HH> \otimes |V3> - |HV> \otimes |H3>) + \frac{b}{2\cdot\sqrt{2}}\cdot(|VH> \otimes |V3> - |VV> \otimes |H3>) \ldots$$
$$+ \left[-\left[\frac{a}{2\cdot\sqrt{2}}\cdot(|VV> \otimes |V3> - |VH> \otimes |H3>) + \frac{b}{2\cdot\sqrt{2}}\cdot(|HV> \otimes |V3> - |HH> \otimes |H3>)\right]\right] \ldots$$
$$+ \frac{a}{2\cdot\sqrt{2}}\cdot(|VV> \otimes |V3> - |VH> \otimes |H3>) + \frac{b}{2\cdot\sqrt{2}}\cdot(|HV> \otimes |V3> - |HH> \otimes |H3>)$$

Kuid avaldis annab ennast teisendada edasi: kui me liidame teistkordselt esimese rea (selle kompenseerimiseks pannes ette kordaja 1/2) ja veel vastandmärkidega 2 rida nagu selles 4-realises avaldises näha, siis on kerge veenduda, et liikmeid rühmitades saab vektori $|123>$ avaldada Belli seisundite kui baasi vektorite kaudu:

$$|123> := \frac{1}{2}\cdot[|t+> \otimes (a\cdot|V3> - b\cdot|H3>) + |t-> \otimes (a\cdot|V3> + b\cdot|H3>) + |s+> \otimes (-a\cdot|H3> + b\cdot|V3>) + |s-> \otimes (-a\cdot|H3> - b\cdot|V3$$

ehk, võttes ümarsulgudes olevate vektorite -- mis näitavad Bobile edastatud abibiti (nr.3) seisundeid -- tähistamiseks tagasi kasutusse 1-veerulised maatriksid ja järjestades liidetavaid ümber, saame lõplikult

$$|123>_{Bellibaasis} := \frac{1}{2}\cdot\left[|s->\otimes\begin{pmatrix}-a\\-b\end{pmatrix}\right]_{1.} + \frac{1}{2}\cdot\left[|s+>\otimes\begin{pmatrix}-a\\b\end{pmatrix}\right]_{2.} + \frac{1}{2}\cdot\left[|t->\otimes\begin{pmatrix}b\\a\end{pmatrix}\right]_{3.} + \frac{1}{2}\cdot\left[|t+>\otimes\begin{pmatrix}-b\\a\end{pmatrix}\right]_{4.} \qquad (4)$$

Kolmikoleku $|123>$ kahest viimasest kirjaviisist paistab välja, et kui olek allutada mõõtmisele nii, et ta kollapseerub tõenäosusega 1/4 üheks 4-st Belli seisundeist, siis kolmas, Bobil olev kvantbitt satub seisundisse, mis on kas seesama, mida tuli teleporteerida, või siis on viimase lihtsal viisil pööratud versioon.

Simuleerime Alice mõõtmist ja saadud numbri 1, 2, 3 või 4 Bobile teatamist läbi suvalise tava-sidekanali või kasvõi läbi ringhäälingu.

$$NeliVrdsetTnst := \begin{pmatrix}\frac{1}{4}\\\frac{1}{4}\\\frac{1}{4}\\\frac{1}{4}\end{pmatrix}$$

TeadeBobile := OmaolekuNr_kuhukukkusOlekmillel(NeliVrdsetTnst)

Seekord:

TeadeBobile = 3

Kahe kvantbiti ühisseisundi mõõtmine Belli analüüsina on oluliselt keerulisem kui ühe kvantbiti mõõtmine. Footonite puhul tuleb polarisatsioonianalüsaatorile lisada kiirekimbujagajaid (poolläbilaskvad peeglid) vms lineaaroptilisi elemente ning ühe asemel ka kaks footonite detektorit.

Mõõtmise tagajärjel teleporteeritav kvantbitt lähtekohas (st Alice juures) hävib. (Tema seisund hävib kasvõi juba selles tähenduses, et põimub Alice juurde jäänud abi-kvantbiti seisundiga).

3. Bob rakendab temale saadetud kvantbitile ühte neljast unitaarpöördest, sõltuvalt Alice'lt saadud teatele.

Bobi tegevuse algoritm teleportatsiooni lõpuleviimiseks seisneb alljärgnevas (temal oleva 3.-nda kvantbitiga).

$$\text{Output}(\text{teade}) := \begin{cases} -I \cdot \begin{pmatrix} -a \\ -b \end{pmatrix} & \text{if } \text{teade} = 1 \\ -\sigma_z \cdot \begin{pmatrix} -a \\ b \end{pmatrix} & \text{if } \text{teade} = 2 \\ \sigma_x \cdot \begin{pmatrix} b \\ a \end{pmatrix} & \text{if } \text{teade} = 3 \\ -\sigma_x \cdot \sigma_z \cdot \begin{pmatrix} -b \\ a \end{pmatrix} & \text{if } \text{teade} = 4 \\ \text{"VIGA!"} & \text{otherwise} \end{cases}$$

Rõhutame, et nagu Alice'gi, ei tea Bob tema kvantbiti seisundit, st *a* ja *b* arvväärtusi.

Pöördeoperaatorite ees siin võib miinusmärgid ka ära jätta. Miks?

Footonite puhul vajalik unitaarpööre teostatakse jällegi lihtsate optikaelementidega -- peeglid, polarisatsioonipöörajad jms.

Bobi unitaarpöördeseadmest väljub kvantbit seisundis, mis on identne Alice juures teleporteerimisele määratud seisundiga -- **seega on kvant-teleportatsioon teoks tehtud.**

Teleportatsiooni simulatsiooni jälgimine

Asume nüüd vanajumalastki võimsamasse positsiooni, millest saame vabalt "vaadata kvantbiti sisse" ja näha, kuidas seisund teleporteerub. **Pane kursor siia avaldisele** SaadaUusFooton ≡ 1 **ja vajuta F9**.

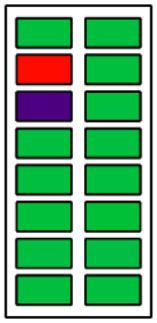
TeadeBobile = 3

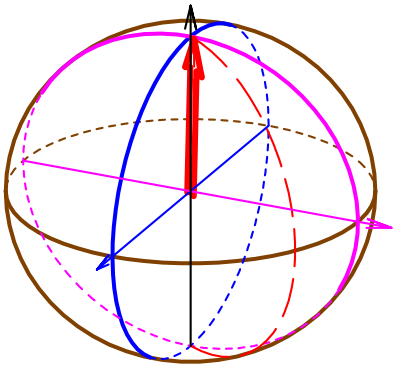
Teleporteeritav seisund (1) oli $\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 \\ 8.88 \times 10^{-3} + 0.02i \end{pmatrix}$ -- väljendatuna komponentides ja Blochi sfääri nurkades

Teleporteeringu tulem: $\theta = 2.01 \cdot \text{deg}$ $\phi = 59.5 \cdot \text{deg}$

$$\text{Output}(\text{TeadeBobile}) = \begin{pmatrix} 1 \\ 8.88 \times 10^{-3} + 0.02i \end{pmatrix}$$

$$|123> = \begin{pmatrix} 0 \\ 0.71 \\ -0.71 \\ 0 \\ 0 \\ 6.28 \times 10^{-3} + 0.01i \\ -6.28 \times 10^{-3} - 0.01i \\ 0 \end{pmatrix}$$





Re Im

|123 >

Seda värvipilti saab vaadata
ka tulpdiagrammina, kui kursoriga
tirida ta soovitava nurga alla.
Lähtenurgad on:
Rotation 0
Tilt 90
Twist 0

Kontrollküsimused.

1. Kas Alice saab oma mõõtmisakti läbi midagi teada teleporteeriva seisundi kohta?
2. Mis tagab selle, et teleportatsiooni käigus ei rikutaks mitteklooni tavuse teoreemi?
3. Missugused kolmikseisundi |123 > kvantbitid on omavahel põimseisundis ja missugused otsekorrutisseisundis?
4. Sama küsimus seisundi kohta pärast Alice tehtud mõõtmist?

Probleemküsimus: kas teleporteeritav kvantbitt võib algselt olla ka põimseisundis mingi muu kvantsüsteemiga?

---------- Töölehe lõpp ----------