Journal of Applied Science and Innovation Studies
ISSN 2998-3606
Vol. 2, No. 4, 2025
Published by STSL Press

Original Paper

Generation of secret keys by the G_3^+ wave functions

Alexander Soiguine

SOiGUINE Supercomputing LLC, Aliso Viejo, CA, USA

Correspondence: Alexander Soiguine, 31 Aurora, Aliso Viejo, CA 92656 USA, info@soiguine.com

Abstract

The superiority of hypothetical supercomputers is not due to faster calculations but due to a different scheme that can be effectively simulated on multithreading computers equipped with the Adreno type of a GPU. In conventional approach the quantum random number generator is implemented through the hardware generating unpredictable random outcomes. In the G_3^+ wave functions approach the random seeding is arbitrary selection of a three-dimensional frame where wave functions are superpositions of the wave functions not changing one component of any bivector observable and those just flipping that component. Thus, infinitely many independent results give truly random outcomes.

Keywords: geometric algebra, wave functions, observables, measurements, GPU, multithreading, OpenCL

1. Introduction

The following perspective is not about generalization of quantum mechanics. It is about a novel theory based on different mathematical structures and different physically meaningful definitions. The scheme suggested in the geometric algebra approach is based on manipulation and transferring of quantum states as operators acting on observables. Wave functions act in that context on static G_3^+ elements through measurements, creating "particles", see (Soiguine, Scattering of Geometric Algebra Wave Functions and Collapse in Measurements, 2020).

Wave functions are identified by points on the $\mathbb{S}^3$ sphere. Measurements of any number of observables by an arbitrary set of wave functions are simultaneously available, see (Soiguine, Scattering of Geometric Algebra Wave Functions and Collapse in Measurements, 2020), (Soiguine, The Geometric Algebra Lift of Qubits and Beyond, 2020), (Soiguine, The Geometric Algebra Structure for Supercomputing, 2024).

States, observables, and values of observations should follow general axioms (Soiguine, The Geometric Algebra Structure for Supercomputing, 2024), p.6:

- Measurement of observable $O(\mu)$ by a state $S(\lambda)$ is a map:

$$\big(S(\lambda), O(\mu)\big) \longrightarrow O(\nu),$$

where $O(\mu)$ is an element of the set of observables. $S(\lambda)$ is element of another set, set of states, though both sets can be formally equivalent.

- The result (value) of a measurement of observable $O(\mu)$ by the state $S(\lambda)$ is a map sequence:

$$\big(S(\lambda), O(\mu)\big) \longrightarrow O(\nu) \longrightarrow V(B),$$

where V is a set of (Boolean) algebra subsets identifying possible results of measurements.

Take an arbitrary wave function, g-qubit, written in some preselected right-hand screw bivector basis $\{B_1, B_2, B_3\}$ with the multiplication rules $B_1B_2 = -B_3, B_1B_3 = B_2, B_2B_3 = -B_1, B_1B_2B_3 = 1$[1] :

[1] Opposite orientation $B_1B_2B_3 = -1$ can be equivalently used

$so(\alpha,\beta,S) = \alpha + \beta_1 B_1 + \beta_2 B_2 + \beta_3 B_3$

Re-write it as

$so(\alpha,\beta,S) = \alpha + \beta_1 B_1 + \beta_2 B_2 + \beta_3 B_3 = \alpha + \beta_1 B_1 + (\beta_3 + \beta_2 B_1) B_3$

Consider its action on an arbitrary observable of the form

$$C = C_1 B_1 + C_2 B_2 + C_3 B_3$$

Probabilities of the results of measurements are measures of wave functions on the $\mathbb{S}^3$ surface giving the considered results, see (Soiguine, The Geometric Algebra Structure for Supercomputing, 2024), pp.50-52.

Measurement of the observable C by the first part of $so(\alpha,\beta,S)$, namely $\alpha + \beta_1 B_1$, does not change the B_1 component of C, C_1. Measurement of C by $\beta_3 + \beta_2 B_1$ just flips that component value giving $-C_1$.

Suppose we are interested in the probability of the result of measurement in which the observable component C_1 does not change. That is the relative measure of wave functions

$\sqrt{\alpha^2 + \beta_1^2}\left(\frac{\alpha}{\sqrt{\alpha^2+\beta_1^2}} + \frac{\beta_1}{\sqrt{\alpha^2+\beta_1^2}} B_1\right)$ in the measurements:

$$\sqrt{\alpha^2 + \beta_1^2}\left(\frac{\alpha}{\sqrt{\alpha^2 + \beta_1^2}} - \frac{\beta_1}{\sqrt{\alpha^2 + \beta_1^2}} B_1\right) C \sqrt{\alpha^2 + \beta_1^2}\left(\frac{\alpha}{\sqrt{\alpha^2 + \beta_1^2}} + \frac{\beta_1}{\sqrt{\alpha^2 + \beta_1^2}} B_1\right)$$

The measure is equal to $\alpha^2 + \beta_1^2$.

Similar calculations give measure $\beta_3^2 + \beta_2^2$ when the component C_1 in the measurement is just flipped.

2. Method

A quantum random number generator creates randomness by observing unpredictable events in nature. It measures quantum behavior that cannot be forecast or repeated.

Every conventional quantum random number generator begins with a quantum process that produces true physical uncertainty. Common examples include light particles and electron tunneling. These sources generate raw randomness at the hardware.

The geometric algebra supercomputing software platform simulates quantum-like processes on standard GPUs, offering effective alternative to traditional quantum computing hardware. It relates to a software-implemented analog supercomputer. Particularly, it means that a quantum process generating raw randomness at the hardware level can be simulated on the geometric algebra supercomputing software platform.

Assume we are measuring any bivector valued observable C by state $\alpha + \beta_1 B_1 + (\beta_3 + \beta_2 B_1) B_3$. Check the sign of the B_1 component of the result of measurement. That is sign of the scalar product $C \cdot B_1$, the latter is equal to minus scalar product of vectors dual to C and B_1, see (Soiguine, The Geometric Algebra Structure for Supercomputing, 2024), pp.13-14.

If the sign of the B_1 component of result of measurement $C \cdot B_1$ is positive, then the generated random result is taken 0. If the sign of the B_1 component of result of measurement $C \cdot B_1$ is negative, then the generated random result is taken 1.

If a sequence of measurements has the length N we get binary string of randomly positioned 0s and 1s, the amount of 0s is $N(\alpha^2 + \beta_1^2)$ and the amount of 1s is $N(\beta_3^2 + \beta_2^2)$.

To get random places of 0s and 1s the measured observable in the sequence of measurements is assumed to have the observable bivector plane randomly oriented[2].

3. Results

The software platform was created as OpenCL program running on Qualcomm Snapdragon ARM computer equipped with Adreno GPU.

Main blocks of the randomly oriented observables identified by vectors normal to the observables plane are the following ones:

- Generate N size array of unit random quaternions

It is implemented, first, by generating three independent random numbers r_1, r_2. r_3 from a uniform distribution between 0 and 1. Then the following transformations are applied to these random numbers to obtain the components of the quaternion:

$$w = \sqrt{1 - r_1} \cdot \sin(2\pi r_2)$$
$$x = \sqrt{1 - r_1} \cdot \cos(2\pi r_2)$$
$$y = \sqrt{r_1} \cdot \sin(2\pi r_3)$$
$$z = \sqrt{r_1} \cdot \cos(2\pi r_3)$$

Thus, the unit quaternion is: $q = (w, x, y, z)$

- Apply the quaternions to initially given observable thus receiving N size array of random observables

- Then OpenCL kernel function executes on Snapdragon ARM Adreno GPU returning the corresponding values of measurements defined in the previous section, the state acting on observables is given.

- After that the kernel function transforms binary elements into decimal which are the required randomly generated numbers which can be used as the secret key in blockchain communications.

4. Discussion

SOiGUINE Supercomputing LLC (https://soiguine.com) is an innovative pre-seed startup developing Geometric Algebra Supercomputing, a software platform that simulates quantum-like computations on standard GPUs, offering a cost-effective alternative to traditional quantum computing hardware.

Every quantum random number generator follows the same sequence of steps: generate a quantum event → measure its outcome → convert it into digital form.

Examples of conventional hardware QNRG include light particles, electron tunneling, or tiny energy fluctuations that exist even in empty space. These sources generate raw randomness at the hardware level before any computation takes place. Detectors record outcomes from that quantum process, such as which path a photon takes, when it arrives, or how its energy changes. Those random events are then converted into binary form. Each detection becomes a 0 or a 1.

Contrary to all that the geometric algebra supercomputing software platform simulates quantum-like processes on standard GPUs, offering effective alternative to traditional quantum computing hardware. It relates to a software-implemented analog supercomputer. Particularly, it means that a quantum process generating raw randomness at the hardware level can be simulated on the geometric algebra supercomputing software platform. The software platform generally relates to high-performance computing systems. More specifically it relates to a software-implemented analog supercomputer

[2] An option exists to fill the N size array with 1s and then put $N(\alpha^2 + \beta_1^2)$ 0s, say, in the beginning of the string and execute some random number of permutations. Disadvantage of this option is that permutations cannot be parallelized by OpenCL kernel.

5. Conclusions

The geometric algebra lift of conventional quantum mechanics qubits is the game-changing quantum leap forward. In this way the new type of quantum computer appeared to be a kind of analog computer keeping and instantly processing information by and on sets of objects possessing an infinite number of degrees of freedom. The approach particularly successfully explained the double slit experiment results (Soiguine, The Geometric Algebra Structure for Supercomputing, 2024) (pp.54-56), the hydrogen atom model (Soiguine, Hydrogen Atom Model in Geometric Algebra Terms, 2023), (Soiguine, The Geometric Algebra Structure for Supercomputing, 2024) (pp.57-60). In the current work we briefly described generating of a sequence of random numbers instead of hardware RNGs.

References

Soiguine, A. (2020). Scattering of Geometric Algebra Wave Functions and Collapse in Measurements. *Journal of Applied Matrhematics and Physics, 8*, 1838-1844.

Soiguine, A. (2020). *The Geometric Algebra Lift of Qubits and Beyond*. LAMBERT Academic Publishing.

Soiguine, A. (2023). Hydrogen Atom Model in Geometric Algebra Terms. *Transnational Journal of Mathematical Analysis and Applications, 11*(1), 29-43.

Soiguine, A. (2024). *The Geometric Algebra Structure for Supercomputing*. LAMBERT Academi Publishing.