http://www.TheoryOfEverything.org

Unimodular rotation of E_8 to H_4 600-cells

J Gregory Moxness*
TheoryOfEverything.org
(Dated: October 19, 2019)

We introduce a unimodular Determinant=1 8×8 rotation matrix to produce four 4 dimensional copies of H_4 600-cells from the 240 vertices of the Split Real Even E_8 Lie group. Unimodularity in the rotation matrix provides for the preservation of the 8 dimensional volume after rotation, which is useful in the application of the matrix in various fields, from theoretical particle physics to 3D visualization algorithm optimization.

PACS numbers: 02.20.-a, 02.10.Yn
Keywords: Coxeter groups, root systems, E8

I. INTRODUCTION

Fig. 1 is the Petrie projection of the largest of the exceptional simple Lie algebras, groups and lattices called E_8. The Split Real Even (SRE) form has 240 vertices and 6720 edges of 8 dimensional (8D) length $\sqrt{2}$. Interestingly, E_8 has been shown to fold to the 4D polychora of H_4 (aka. the 120 vertex 720 edge 600-cell) and a scaled copy $H_4\Phi$[1][2], where $\Phi = \frac{1}{2}\left(1+\sqrt{5}\right) = 1.618...$ is the big golden ratio and $\varphi = \frac{1}{2}\left(\sqrt{5}-1\right) = 1/\Phi = \Phi - 1 = 0.618...$ is the small golden ratio.

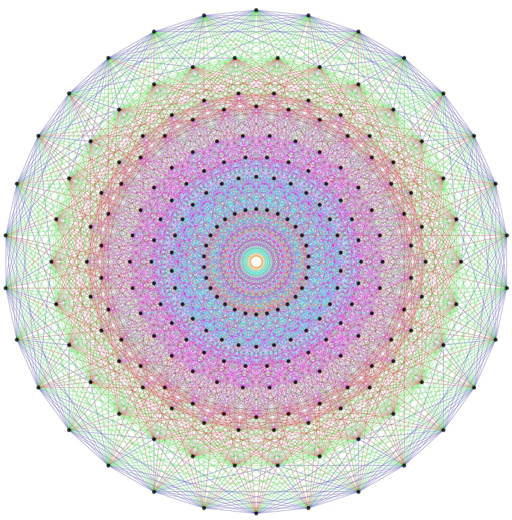

FIG. 1: E_8 Petrie projection

In my previous papers on the topic [3][4], a specific matrix for performing the rotation of the SRE E_8 group of root vertices to the vertices of H_4 (a.k.a. the 600-cell) was shown to be that of (1).

$$\mathrm{H4_{fold}} = \begin{pmatrix} \varphi^2 & 0 & 0 & 0 & \Phi & 0 & 0 & 0 \\ 0 & -\varphi & 1 & 0 & 0 & \varphi & 1 & 0 \\ 0 & 1 & 0 & -\varphi & 0 & 1 & 0 & \varphi \\ 0 & 0 & -\varphi & 1 & 0 & 0 & \varphi & 1 \\ \Phi & 0 & 0 & 0 & \varphi^2 & 0 & 0 & 0 \\ 0 & \varphi & 1 & 0 & 0 & -\varphi & 1 & 0 \\ 0 & 1 & 0 & \varphi & 0 & 1 & 0 & -\varphi \\ 0 & 0 & \varphi & 1 & 0 & 0 & -\varphi & 1 \end{pmatrix} \quad (1)$$

The convex hull of two opposite edges of a regular icosahedron forms a golden rectangle (as shown in Fig. 2). The twelve vertices of the icosahedron can be decomposed in this way into three mutually-perpendicular golden rectangles, whose boundaries are linked in the pattern of the Borromean rings. Columns 2-4 of $\mathrm{H4_{fold}}$ contains 6 of the 12 vertices of this icosahedron, including 2 at the origin (with the other 6 of 12 icosahedron vertices being the reflection of these through the origin).

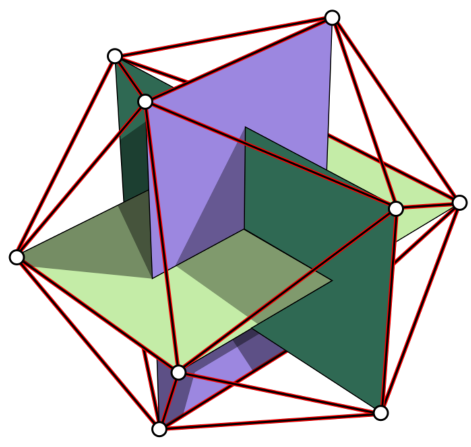

FIG. 2: The Icosahedron formed from 3 mutually-perpendicular golden rectangles

The trace of this matrix is $2(\phi^2 - \phi + 1) = 1.527$ and its determinant $Det = (2\sqrt{\phi})^8 = 37.349$.

*URL: http://www.TheoryOfEverything.org/TOE/JGM; mailto:jgmoxness@TheoryOfEverything.org

Notice that $\mathrm{H4}_{\mathrm{fold}} = \mathrm{H4}^T_{\mathrm{fold}}$ such that it is symmetric with a quaternion-octonion Cayley-Dickson-like structure.

Only the first 4 rows are needed for folding E_8 to H_4 by dot product with each vertex. This results in two copies of H_4 scaled by Φ. Using the full matrix to rotate E_8 results in not two, but four copies of H_4 600-cell with the left (L) 4 dimensions associated with the two scaled copies (H_4 and $H_4\Phi$) and the right (R) 4 dimensions associated with another two copies (H_4 and $H_4\Phi$). Rotation back to E_8 is achieved with a rotation matrix of $\mathrm{H4}^{-1}_{\mathrm{fold}}$.

II. THE UNIMODULARITY FACTOR

The Platonic solid icosahedral symmetry establishes some valuable utility in this particular construction of $\mathrm{H4}_{\mathrm{fold}}$. Yet, the non-unimodularity of the determinant causes the resulting 8D volume of the objects involved in a rotation (or projection) between $E_8 \leftrightarrow H_4$ to vary. In order to correct this, while keeping the general structure of the matrix the same, we simply divide the matrix by a factor of $2\sqrt{\phi}$, giving a $Det = 1$. This gives:

$\mathrm{H4}_{\mathrm{uni}} =$

$$\begin{pmatrix} \sqrt{\varphi^3} & 0 & 0 & 0 & \frac{1}{\sqrt{\varphi^3}} & 0 & 0 & 0 \\ 0 & -\sqrt{\varphi} & \frac{1}{\sqrt{\varphi}} & 0 & 0 & \sqrt{\varphi} & \frac{1}{\sqrt{\varphi}} & 0 \\ 0 & \frac{1}{\sqrt{\varphi}} & 0 & -\sqrt{\varphi} & 0 & \frac{1}{\sqrt{\varphi}} & 0 & \sqrt{\varphi} \\ 0 & 0 & -\sqrt{\varphi} & \frac{1}{\sqrt{\varphi}} & 0 & 0 & \sqrt{\varphi} & \frac{1}{\sqrt{\varphi}} \\ \frac{1}{\sqrt{\varphi^3}} & 0 & 0 & 0 & \sqrt{\varphi^3} & 0 & 0 & 0 \\ 0 & \sqrt{\varphi} & \frac{1}{\sqrt{\varphi}} & 0 & 0 & -\sqrt{\varphi} & \frac{1}{\sqrt{\varphi}} & 0 \\ 0 & \frac{1}{\sqrt{\varphi}} & 0 & \sqrt{\varphi} & 0 & \frac{1}{\sqrt{\varphi}} & 0 & -\sqrt{\varphi} \\ 0 & 0 & \sqrt{\varphi} & \frac{1}{\sqrt{\varphi}} & 0 & 0 & -\sqrt{\varphi} & \frac{1}{\sqrt{\varphi}} \end{pmatrix} /2 \tag{2}$$

III. $\mathrm{H4}_{\mathrm{fold}}$ FROM 2 QUBIT QUANTUM COMPUTING CNOT AND SWAP GATES

Looking at the four quadrants of $\mathrm{H4}_{\mathrm{fold}}$ and $\mathrm{H4}_{\mathrm{uni}}$, we see that they resemble a combination of the unitary Hermitian matrices commonly used for Quantum Computing (QC) qubit logic, namely those of the 2 qubit CNOT (3) and SWAP (4) gates. Taking these patterns, combined with the recursive functions that build Φ from the Fibonacci sequence, it is straightforward to derive both $\mathrm{H4}_{\mathrm{fold}}$ and $\mathrm{H4}_{\mathrm{uni}}$ from scaled QC logic gates.

$$\mathrm{CNOT}= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{3}$$

$$\mathrm{SWAP}= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4}$$

The code to establish CNOT and SWAP implementations of $\mathrm{H4}_{\mathrm{fold}}$ is naively done (in *Mathematica*™ code) as shown in Fig. 3.

```
SWAP + # CNOT & /@ {-φ, φ };
Flatten /@ Transpose@Join[
    {Flatten[%, 1]},
    {Flatten[Reverse@%, 1]}]
```

$$\begin{pmatrix} 1-\varphi & 0 & 0 & 0 & \varphi+1 & 0 & 0 & 0 \\ 0 & -\varphi & 1 & 0 & 0 & \varphi & 1 & 0 \\ 0 & 1 & 0 & -\varphi & 0 & 1 & 0 & \varphi \\ 0 & 0 & -\varphi & 1 & 0 & 0 & \varphi & 1 \\ \varphi+1 & 0 & 0 & 0 & 1-\varphi & 0 & 0 & 0 \\ 0 & \varphi & 1 & 0 & 0 & -\varphi & 1 & 0 \\ 0 & 1 & 0 & \varphi & 0 & 1 & 0 & -\varphi \\ 0 & 0 & \varphi & 1 & 0 & 0 & -\varphi & 1 \end{pmatrix}$$

FIG. 3: Producing $\mathrm{H4}_{\mathrm{fold}}$ from 2 Qubit CNOT and SWAP QC Gates

More interestingly, we can produce a similar result using a recursive function for Φ using the Fibonacci sequence. This is shown in Figs. 5-6 in Appendix A, where we iterate the *Mathematica*™ Fibonacci function $n = 10$ times. As $n \to \infty$, the matrix resolves to $\mathrm{H4}_{\mathrm{fold}}$ or $\mathrm{H4}_{\mathrm{uni}}$. The numerical result for the first 4 rows of $\mathrm{H4}_{\mathrm{fold}}$ is shown in Fig. 4 at $n = 10$.

```
rndMat@mat
```

$$\begin{pmatrix} 0.382 & 0. & 0. & 0. & 1.618 & 0. & 0. & 0. \\ 0. & -0.618 & 1. & 0. & 0. & 0.618 & 1. & 0. \\ 0. & 1. & 0. & -0.618 & 0. & 1. & 0. & 0.618 \\ 0. & 0. & -0.618 & 1. & 0. & 0. & 0.618 & 1. \end{pmatrix}$$

FIG. 4: Numerical result for the first 4 rows of $\mathrm{H4}_{\mathrm{fold}}$ from the 2 Qubit CNOT and SWAP QC gates and an integer Fibonacci series function output after $n = 10$ iterations

IV. CONCLUSION

Instead of simply folding the 8D E_8 vertices into 4D pairs of H_4 and $H_4\Phi$ vertices, we rotate them using an 8×8 matrix. This transforms E_8 into a fourfold H_4 600-cell structure. We show that bringing unimodularity to the folding matrix with a $Det = 1$ is a simple modification. We also show that the folding matrix can easily be generated using 2 qubit QC matrices and recursive functions related to the Fibonacci sequence.

Acknowledgments

I would like to thank my wife for her love and patience and those in academia who have taken the time to review this work.

[1] M. Koca and N. Koca, Turkish Journal of Physics **22**, 421 (1998).
[2] D. A. Richter, ArXiv e-prints math.GM/0704.3091 (2007), 0704.3091.
[3] J. G. Moxness, `www.vixra.org/abs/1411.0130` (2014).
[4] J. G. Moxness, `www.vixra.org/abs/1804.0065` (2018).

Appendix A: *Mathematica*™ code

Producing the first 4 rows of $H4_{\text{fold}}$ from 2 Qubit CNOT and SWAP QC gates and an integer Fibonacci series function

```
fb = Fibonacci;
im = IdentityMatrix;
nC@0 := CNOT.SWAP;
nC@1 := nC[0]ᵀ;
nC@i_ := nC[i - 2] + nC[i - 1];
nCInv := Inverse@nC[#]ᵀ &;
{mat = Join[
    fb[# + 1]/fb[#] (2 nC[# - 1].nCInv@# - im@4)ᵀ⟦{1, 4, 3, 2}⟧,
    fb[# - 1]/fb[# + 1] (2 nC[# + 1].nCInv@# + nC@1)ᵀ⟦{1, 3, 2, 4}⟧]ᵀ
  } & /@ Range@10
```

FIG. 5: Mathematica Code

$$
\left(\begin{array}{c}
\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\
\begin{pmatrix} 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -2 & 2 & 0 & 0 & \frac{1}{2} & \frac{3}{2} & 0 \\ 0 & 2 & 0 & -2 & 0 & \frac{3}{2} & 0 & \frac{1}{2} \\ 0 & 0 & -2 & 2 & 0 & 0 & \frac{1}{2} & \frac{3}{2} \end{pmatrix} \\
\begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 & \frac{13}{9} & 0 & 0 & 0 \\ 0 & -\frac{1}{3} & \frac{2}{3} & \frac{1}{6} & 0 & \frac{16}{27} & \frac{22}{27} & \frac{1}{27} \\ 0 & \frac{2}{3} & \frac{1}{6} & -\frac{1}{3} & 0 & \frac{22}{27} & \frac{1}{27} & \frac{16}{27} \\ 0 & \frac{1}{6} & -\frac{1}{3} & \frac{2}{3} & 0 & \frac{1}{27} & \frac{16}{27} & \frac{22}{27} \end{pmatrix} \\
\begin{pmatrix} \frac{1}{3} & 0 & 0 & 0 & \frac{42}{25} & 0 & 0 & 0 \\ 0 & -\frac{16}{21} & \frac{8}{7} & -\frac{1}{21} & 0 & \frac{108}{175} & \frac{188}{175} & -\frac{2}{175} \\ 0 & \frac{8}{7} & -\frac{1}{21} & -\frac{16}{21} & 0 & \frac{188}{175} & -\frac{2}{175} & \frac{108}{175} \\ 0 & -\frac{1}{21} & -\frac{16}{21} & \frac{8}{7} & 0 & -\frac{2}{175} & \frac{108}{175} & \frac{188}{175} \end{pmatrix} \\
\begin{pmatrix} \frac{2}{5} & 0 & 0 & 0 & \frac{51}{32} & 0 & 0 & 0 \\ 0 & -\frac{54}{95} & \frac{18}{19} & \frac{2}{95} & 0 & \frac{375}{608} & \frac{591}{608} & \frac{3}{608} \\ 0 & \frac{18}{19} & \frac{2}{95} & -\frac{54}{95} & 0 & \frac{591}{608} & \frac{3}{608} & \frac{375}{608} \\ 0 & \frac{2}{95} & -\frac{54}{95} & \frac{18}{19} & 0 & \frac{3}{608} & \frac{375}{608} & \frac{591}{608} \end{pmatrix} \\
\begin{pmatrix} \frac{3}{8} & 0 & 0 & 0 & \frac{275}{169} & 0 & 0 & 0 \\ 0 & -\frac{125}{196} & \frac{50}{49} & -\frac{3}{392} & 0 & \frac{5120}{8281} & \frac{8370}{8281} & -\frac{15}{8281} \\ 0 & \frac{50}{49} & -\frac{3}{392} & -\frac{125}{196} & 0 & \frac{8370}{8281} & -\frac{15}{8281} & \frac{5120}{8281} \\ 0 & -\frac{3}{392} & -\frac{125}{196} & \frac{50}{49} & 0 & -\frac{15}{8281} & \frac{5120}{8281} & \frac{8370}{8281} \end{pmatrix} \\
\begin{pmatrix} \frac{5}{13} & 0 & 0 & 0 & \frac{712}{441} & 0 & 0 & 0 \\ 0 & -\frac{1024}{1677} & \frac{128}{129} & \frac{5}{1677} & 0 & \frac{35152}{56889} & \frac{56656}{56889} & \frac{40}{56889} \\ 0 & \frac{128}{129} & \frac{5}{1677} & -\frac{1024}{1677} & 0 & \frac{56656}{56889} & \frac{40}{56889} & \frac{35152}{56889} \\ 0 & \frac{5}{1677} & -\frac{1024}{1677} & \frac{128}{129} & 0 & \frac{40}{56889} & \frac{35152}{56889} & \frac{56656}{56889} \end{pmatrix} \\
\begin{pmatrix} \frac{8}{21} & 0 & 0 & 0 & \frac{468}{289} & 0 & 0 & 0 \\ 0 & -\frac{4394}{7077} & \frac{338}{337} & -\frac{8}{7077} & 0 & \frac{120393}{194786} & \frac{195091}{194786} & -\frac{26}{97393} \\ 0 & \frac{338}{337} & -\frac{8}{7077} & -\frac{4394}{7077} & 0 & \frac{195091}{194786} & -\frac{26}{97393} & \frac{120393}{194786} \\ 0 & -\frac{8}{7077} & -\frac{4394}{7077} & \frac{338}{337} & 0 & -\frac{26}{97393} & \frac{120393}{194786} & \frac{195091}{194786} \end{pmatrix} \\
\begin{pmatrix} \frac{13}{34} & 0 & 0 & 0 & \frac{4893}{3025} & 0 & 0 & 0 \\ 0 & -\frac{9261}{15011} & \frac{882}{883} & \frac{13}{30022} & 0 & \frac{1650768}{2671075} & \frac{2669478}{2671075} & \frac{273}{2671075} \\ 0 & \frac{882}{883} & \frac{13}{30022} & -\frac{9261}{15011} & 0 & \frac{2669478}{2671075} & \frac{273}{2671075} & \frac{1650768}{2671075} \\ 0 & \frac{13}{30022} & -\frac{9261}{15011} & \frac{882}{883} & 0 & \frac{273}{2671075} & \frac{1650768}{2671075} & \frac{2669478}{2671075} \end{pmatrix} \\
\begin{pmatrix} \frac{21}{55} & 0 & 0 & 0 & \frac{12818}{7921} & 0 & 0 & 0 \\ 0 & -\frac{78608}{127105} & \frac{2312}{2311} & -\frac{21}{127105} & 0 & \frac{11313500}{18305431} & \frac{18309612}{18305431} & -\frac{714}{18305431} \\ 0 & \frac{2312}{2311} & -\frac{21}{127105} & -\frac{78608}{127105} & 0 & \frac{18309612}{18305431} & -\frac{714}{18305431} & \frac{11313500}{18305431} \\ 0 & -\frac{21}{127105} & -\frac{78608}{127105} & \frac{2312}{2311} & 0 & -\frac{714}{18305431} & \frac{11313500}{18305431} & \frac{18309612}{18305431} \end{pmatrix}
\end{array}\right)
$$

FIG. 6: Integer Fibonacci series function output for each of $n = 10$ iterations