

The Isomorphism of H_4 and E_8

J Gregory Moxness*
TheoryOfEverything.org
 (Dated: October 30,2023)

This paper gives an explicit isomorphic mapping from the 240 real \mathbb{R}^8 roots of the E_8 Gosset 4_{21} 8-polytope to two golden ratio scaled copies of the 120 root H_4 600-cell quaternion 4-polytope using a traceless 8×8 rotation matrix \mathbb{U} with palindromic characteristic polynomial coefficients and a unitary form $e^{i\mathbb{U}}$. It also shows the inverse map from a single H_4 600-cell to E_8 using a $4\text{D} \leftrightarrow 8\text{D}$ chiral left \leftrightarrow right mapping function, φ scaling, and \mathbb{U}^{-1} . This approach shows that there are actually four copies of each 600-cell living within E_8 in the form of chiral $H_{4L} \oplus \varphi H_{4L} \oplus H_{4R} \oplus \varphi H_{4R}$ roots. In addition, it demonstrates a quaternion Weyl orbit construction of H_4 -based 4-polytopes that provides an explicit mapping between E_8 and four copies of the tri-rectified Coxeter-Dynkin diagram of H_4 , namely the 120-cell of order 600. Taking advantage of this property promises to open the door to as yet unexplored E_8 -based Grand Unified Theories or GUTs.

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

I. INTRODUCTION

Fig. 1 is the Petrie projection of the Gosset 4_{21} 8-polytope derived from the Split Real Even (SRE) form of the E_8 Lie group with unimodular lattice in \mathbb{R}^8 . It has 240 vertices and 6,720 edges of 8-dimensional (8D) length $\sqrt{2}$. E_8 is the largest of the exceptional simple Lie algebras, groups, lattices, and polytopes related to octonions (\mathbb{O}), (8,4) Hamming codes, and 3-qubit (8 basis state) Hadamard matrix gates. An important and related higher dimensional structure is the \mathbb{R}^{24} (\mathbb{C}^{12}) Leech lattice ($\Lambda_{24} \supset E_8 \oplus E_8 \oplus E_8$), with its binary (ternary) Golay code construction.

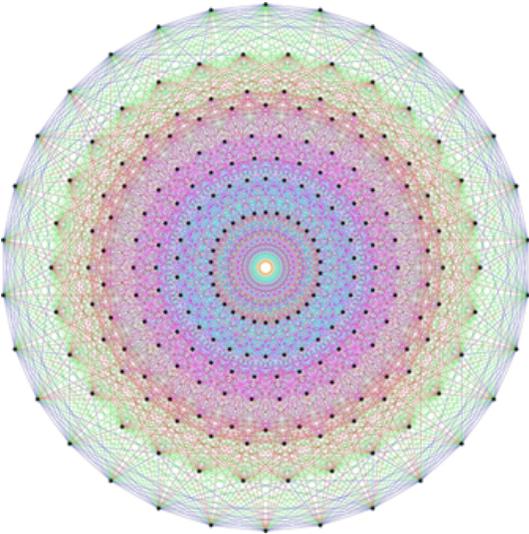


FIG. 1. E_8 4_{21} Petrie projection

It is widely known [1]-[14] that the E_8 can be projected, mapped, or "folded" (as shown in Fig. 2) to two golden ratio $\varphi = \frac{1}{2}(1 + \sqrt{5}) \approx 1.618$ scaled copies of the 4 dimensional 120 vertex 720 edge H_4 600-cell. Folding an 8D object into a 4D one can be done by projecting each vertex using its dot product with a 4×8 matrix[11]. This produces $H_4 \oplus \varphi H_4$, where H_4 is the binary icosahedral group $2I$ of order 120, a subgroup of $\text{Spin}(3)$. It covers H_3 as the full icosahedral group I_h of order 120, a subgroup of $\text{SO}(3)$. The binary icosahedral group is the double cover of the alternating group A_5 .

Despite others'[2][9] recent attempts, the inverse morphism or "unfolding" from H_4 to E_8 is less trivial given that the matrix is not square and lacks an inverse. Yet, a real (\mathbb{R}) symmetric volume preserving $\text{Det}(\mathbb{U})=1$ rotation matrix(1) was derived in 2012 and documented[11][12][13]. The quadrant structure of \mathbb{U} rotates E_8 into four 4D copies of H_4 600-cells, with the original two (L)eft and (R)ight side unit scaled 4D copies related to the two L/R φ scaled copies which we now identify as $H_4(L \oplus R \oplus 1 \oplus \varphi)$. This traceless form of \mathbb{U} has palindromic characteristic coefficients and provides for an explicit isomorphic mapping of $E_8 \leftrightarrow H_4(L \oplus R \oplus 1 \oplus \varphi)$. This involves using a bidirectional L \leftrightarrow R mapping function (mapLR) and \mathbb{U}^{-1} (2). The process is described and visualized in Section II. It is interesting to note the exchange of $1 \leftrightarrow \varphi$ in $\mathbb{U} \leftrightarrow \mathbb{U}^{-1}$, excluding $-\varphi^2$.

$$\mathbb{U} = \begin{pmatrix} 1 - \varphi & 0 & 0 & 0 & 0 & 0 & 0 & -\varphi^2 \\ 0 & -1 & \varphi & 0 & 0 & \varphi & 1 & 0 \\ 0 & \varphi & 0 & 1 & -1 & 0 & \varphi & 0 \\ 0 & 0 & -1 & \varphi & \varphi & 1 & 0 & 0 \\ 0 & 0 & 1 & \varphi & \varphi & -1 & 0 & 0 \\ 0 & \varphi & 0 & 1 & -1 & 0 & \varphi & 0 \\ 0 & 1 & \varphi & 0 & 0 & \varphi & -1 & 0 \\ -\varphi^2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 - \varphi \end{pmatrix} / (2\sqrt{\varphi}) \quad (1)$$

* <https://www.TheoryOfEverything.org/theToE>;
 mailto:jgmoxness@TheoryOfEverything.org

$$U^{-1} = \frac{1}{(2\sqrt{\varphi})} \begin{pmatrix} \varphi - 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\varphi^2 \\ 0 & -\varphi & 1 & 0 & 0 & 1 & \varphi & 0 & 0 \\ 0 & 1 & 0 & \varphi & -\varphi & 0 & 1 & 0 & 0 \\ 0 & 0 & -\varphi & 1 & 1 & \varphi & 0 & 0 & 0 \\ 0 & 0 & \varphi & 1 & 1 & -\varphi & 0 & 0 & 0 \\ 0 & 1 & 0 & \varphi & -\varphi & 0 & 1 & 0 & 0 \\ 0 & \varphi & 1 & 0 & 0 & 1 & -\varphi & 0 & 0 \\ -\varphi^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \varphi - 1 \end{pmatrix} \quad (2)$$

A. Generating Polytopes

The quaternion (\mathbb{H}) Weyl group orbit $O(\Lambda) = W(H_4) = I$ of order 120 is constructed from the parent orbit (1000) of the Coxeter-Dynkin diagram for H_4 shown in Fig. 2b. This results in the 600-cell 4-polytope of order 120 labeled here and in [3] as I. In addition, U provides for a direct mapping from E_8 to four $L \oplus R \oplus 1 \oplus \varphi$ copies of the tri-rectified parent of H_4 (i.e. the filled node 1 is shifted right 3 times giving 0001), which is the 120-cell of order 600 labeled here and in [3] as J. Both of these 4-polytopes are shown in Appendix A Figs. 14-16. The detail of the quaternion Weyl orbit construction is described in Section III.

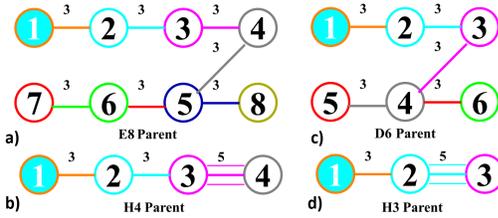


FIG. 2. a) E_8 Dynkin diagram in folding orientation
b) The associated Coxeter-Dynkin diagram of H_4
c) D_6 Dynkin diagram in folding orientation
d) The associated Coxeter-Dynkin diagram of H_3

In addition to the 240 root 4_{21} E_8 8-polytope identified by its Coxeter-Dynkin diagram in Fig. 3a, there are 2^8 possible orbits using only 0's \leftrightarrow 1's, empty \leftrightarrow filled, or ringed nodes of the E_8 Coxeter-Dynkin diagram, including the snub (00000000) orbit. Several other orbit permutations are commonly represented visually using the Petrie projection basis. They are the 2,160 root 2_{41} and 17,280 root 1_{42} 8-polytopes, which are constructed by generating the resulting roots by moving the filled (or ringed) node to each of the two other ends of the Dynkin diagram, as shown in Figs. 3b and 3c respectively.

B. 8D Platonic Rotation

Interestingly from [13], U can be generated using a combination of the unimodular matrices commonly used

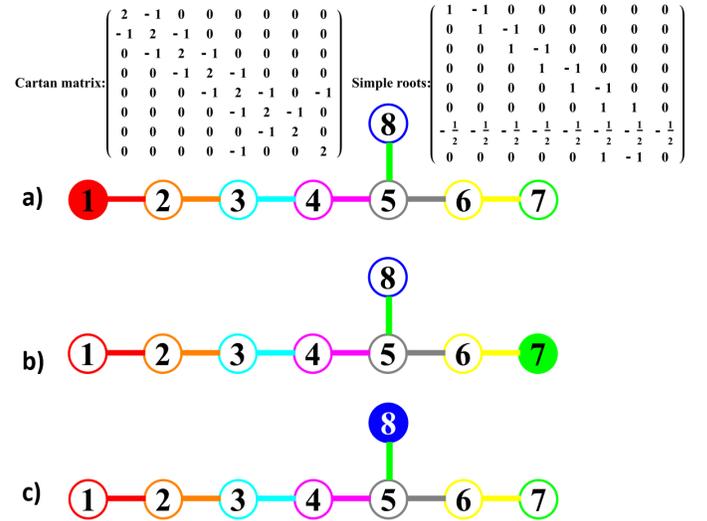


FIG. 3. E_8 Dynkin diagrams a) 4_{21} , b) 2_{41} , c) 1_{42}

Also shown are the Cartan and simple root matrices which correspond to the common Coxeter-Dynkin representation of the diagrams

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 U from scaled QC logic gates.[14]

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (3)$$

$$\text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4)$$

C. 2D and 3D Projection

Projection of E_8 to 2D (or 3D) requires 2 (or 3) basis vectors $\{X, Y, Z\}$. For the Petrie projection shown in Fig. 1, we start with the basis vectors in (5), which are simply the two 2D Petrie projection basis vectors of the 600-cell (a.k.a. the Van Oss projection), with an optional 3rd (z) basis vector added for an interesting 3D projection[11].

$$\begin{aligned} x &= \{ 0, \varphi 2 \sin \frac{2\pi}{15}, 2 \sin \frac{2\pi}{15}, 0, 0, 0, 0, 0 \} \\ y &= \{ -\varphi 2 \sin \frac{2\pi}{30}, 0, 0, 1, 0, 0, 0, 0 \} \\ z &= \{ 1, 0, 0, 0, \varphi 2 \sin \frac{2\pi}{30}, 0, 0, 0, 0 \} \end{aligned} \quad (5)$$

$\{X, Y, Z\} = \mathbb{U}.\{x, y, z\}$ as shown in (6).

$$\begin{aligned} X &= \{0 \quad .252 \quad .427 \quad -.319 \quad .319 \quad .427 \quad .781 \quad 0\} \\ Y &= \{.0821 \quad 0 \quad -.393 \quad .636 \quad .636 \quad .393 \quad 0 \quad .348\} \\ Z &= \{-.242 \quad 0 \quad -.132 \quad .215 \quad .215 \quad .132 \quad 0 \quad -1.03\} \end{aligned} \quad (6)$$

D. 3D Platonic Solid Projection

This basis is derived from the icosahedral symmetry of the H_3 -based Platonic solid. The twelve vertices of the icosahedron can be decomposed into three mutually-perpendicular golden rectangles (as shown in Fig. 4), whose boundaries are linked in the pattern of the Borromean rings. Rows (or columns) 2-4 (or 5-8) of \mathbb{U} contain 6 of the 12 vertices of this icosahedron, including 2 at the origin with the other 6 of 12 icosahedron vertices being the antipodal reflection of these through the origin. These 2 (or 3) rows can then be used as a kind of ‘‘Platonic solid projection prism’’ to form the 2 (or 3) 8D basis vectors used in the 2D (or 3D) projection of 4_{21} , 2_{41} , and 1_{42} .

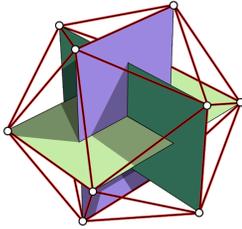


FIG. 4. The icosahedron formed from 3 mutually-perpendicular golden rectangles

Orthogonal projection to 3D after \mathbb{U} folding (i.e. selecting one of 56 unique subsets of any 3 dimensions, here we use $\{1, 2, 3\}$) manifests a large number of concentric hulls with Platonic and Archimedean solid related structures. The eight projected 3D hulls of 4_{21} include two φ scaled sets of four hulls from two 600-cells ($H_4 \oplus \varphi H_4$) as shown in Appendix A Fig. 14. 2_{41} and 1_{42} projections of E_8 are shown in Figs. 5-6.

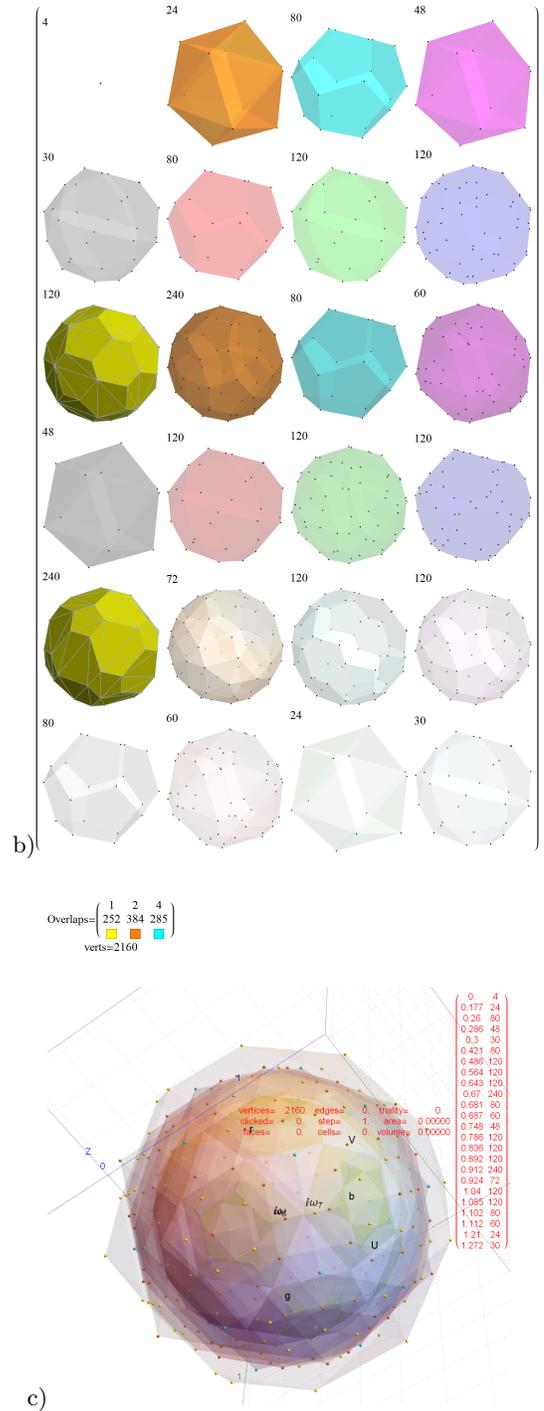
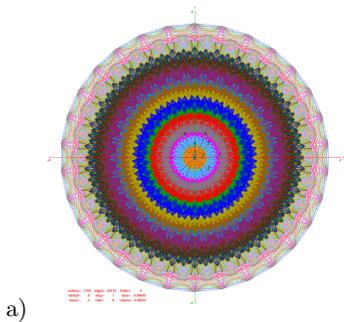


FIG. 5. 2_{41} projections of its 2,160 vertices
a) 2D to the E_8 Petrie projection using basis vectors X and Y from (6) with 8-polytope radius $2\sqrt{2}$ and 69,120 edges of length $\sqrt{2}$.
b) 3D projections with vertices sorted and tallied by their 3D norm generating the increasingly transparent hulls for each set of tallied norms. Notice the last two outer hulls are a combination of two overlapped Icosahedrons (24) and a Icosidodecahedron (30).
c) Combined 3D hulls with the overlapping vertices color coded by overlap count. Also shown is a list (in red) of the normed hull distance and the number of vertices in the group.

II. THE PALINDROMIC UNITARY MATRIX

The particular maximal embedding of E_8 at height 248 that we are interested in for this work is shown in Appendix C Fig. 19 as the special orthogonal group of $SO(16)=D_8$ at height $(120=112+4+4)+128'$, where 112 is interpreted as the subgroup embeddings of $SO(8)\otimes SO(8)=D_4\otimes D_4$ and $128'$ is interpreted as symplectic subgroup embeddings of C_8 where $Sp(8)\otimes Sp(8)=C_4\otimes C_4$ at height $136=128+4+4$. These selected embeddings correspond to the 112 integer D_8 vertices and the 128 half-integer BC_8 vertices given by SRE E_8 , in addition to the $8\oplus 8$ generator roots for a total of 2^8 . This is in 1::1 correspondence with the canonical root vertex ordering from the 9th row of the palindromic Pascal triangle $\{1, 8, 28, 56, 35\overline{35}, \overline{56}, \overline{28}, \overline{8}, \overline{1}\}$, where each entry in the list gives the number of vertices that alternate between half-integer BC_8 and integer D_8 vertex sets, with the right 5 overbar sets of 128 vertices being the negated vertices of the left 5 sets of 128 in reverse order.

These embeddings have an isomorphic connection to \mathbb{U} and provide the $E_8 \leftrightarrow H_4(L\oplus R\oplus 1\oplus \varphi)$ mapping via `mapLR`. The *Mathematica*TM code for `mapLR` and the code to validate the $E_8 \leftrightarrow H_4$ isomorphism is shown in Appendix D Fig. 21. It demonstrates that E_8 rotates into four 4D copies of H_4 600-cells, with the original two (L)eft side φ scaled 4D copies related to the two (R)ight side unscaled 4D copies. `testtest` Due to the palindromic structure of \mathbb{U} , the H_{4L} and H_{4R} are also palindromic with each R vertex being the reverse order of the L vertex, along with `mapLR` exchanges in the (S)ub 24-cell vertices. For each L vertex that is not a member of the (T)etrahedral group's self-dual D_4 24-cell (or φT), the R vertex will be a member of the scaled φS (or S) respectively. This is due to the exchange of $\varphi^{3/2} \leftrightarrow \varphi^{-3/2}$ in `mapLR` which changes the norm (i.e. to/from a small norm= $1/\sqrt{\varphi}$ or a large norm= $\sqrt{\varphi}$). The 24-cell T vertices are unaffected by `mapLR` exchange and have L and R vertex values of the same norm and palindromic opposite entries, with the larger φH_4 having the same signs and the smaller unit scaled H_4 having opposite signs.

It is clear that \mathbb{U} is traceless, but it is not unitary. Since \mathbb{U} is Hermitian, it is easily made unitary as $e^{i\mathbb{U}}$. While that is unitary it is not traceless, so it is not an A_7 group $SU(8)$ symmetry. For the identification of their palindromic characteristic polynomial coefficients, see Figs. 7-8.

See Appendix D Figs. 22-23 showing the detail of the $E_8 \leftrightarrow H_4(L\oplus R\oplus 1\oplus \varphi)$ isomorphism and the patterns within their respective vertex roots.

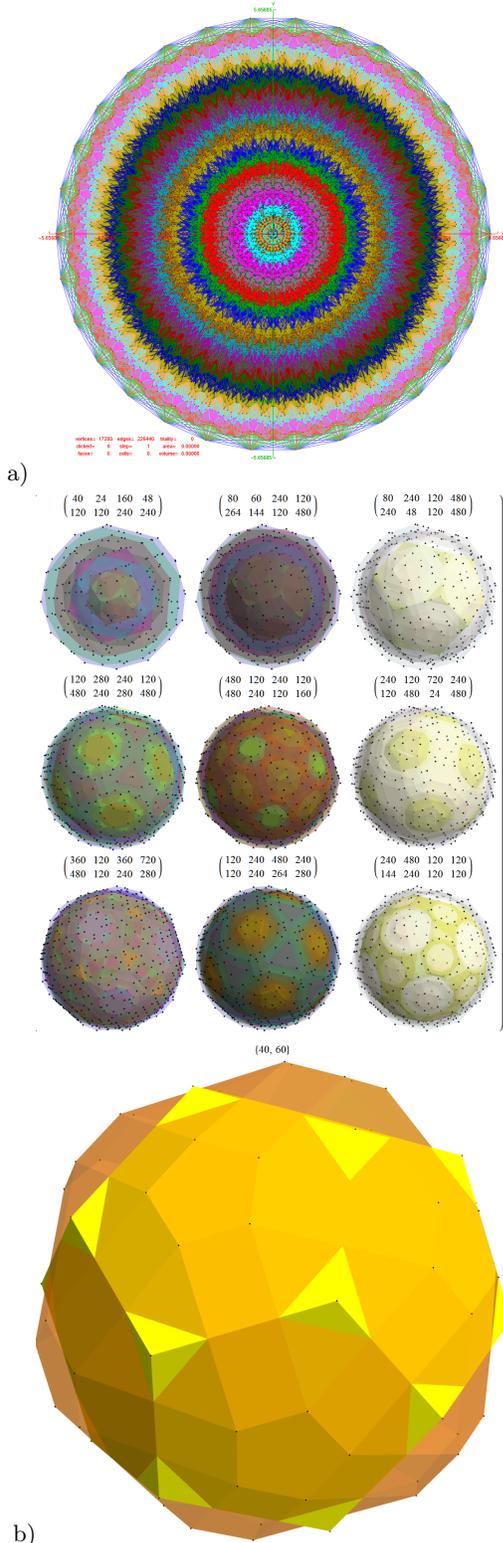


FIG. 6. 142 projections of its 17,280 vertices
a) 2D to the E_8 Petrie projection using basis vectors X and Y from (6) with 8-polytope radius $4\sqrt{2}$ and 483,840 edges of length $\sqrt{2}$ (with 53% of inner edges culled for display clarity).
b) 3D projections with vertices sorted and tallied by their 3D norm generating the increasingly transparent hulls for each set of tallied norms. Notice the last two outer hulls are a combination of two overlapped Dodecahedra (40) and a irregular Rhombicosidodecahedron (60).

```
(* Show the Determinant of U=1 *)
Det@U
N[% /. φRep]
Out[*]=  $\frac{64 \varphi^9 - 64 \varphi^3}{256 \varphi^6}$ 
Out[*]= 1.

(* Show the Trace of U=0 *)
octSimplify /@Tr@U
Chop@N[% /. φRep]
Out[*]=  $-\frac{1}{\varphi^{3/2}} - \sqrt{\frac{1}{\varphi}} + \sqrt{\varphi}$ 
Out[*]= 0

In[*]:= (* Show the Eigensystem of U *)
octSimplify /@FullSimplify[Eigenvalues@U,
Assumptions -> φAssumptions]
Total@N[% /. φRep]
FullSimplify[Eigenvectors@U, Assumptions -> φAssumptions]
Out[*]=  $\left\{ -\sqrt{\frac{1}{\varphi}}, -\sqrt{\frac{1}{\varphi}}, \sqrt{\frac{1}{\varphi}}, -\sqrt{\varphi}, \sqrt{\varphi}, \sqrt{\varphi}, -\sqrt{\varphi}, \sqrt{\frac{1}{\varphi}} \right\}$ 
Out[*]= 0.


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


In[*]:= (* Get the Characteristic coefficients *)
FullSimplify[CharacteristicPolynomial[U, x],
Assumptions -> φAssumptions]
Out[*]=  $\frac{1}{\varphi^{9/2}} (\varphi^3 x^7 (\varphi (1. - 1. \varphi) + 1.) + \varphi^{3/2} (0.25 \varphi^6 - 0.25) +$ 
 $1. \varphi^{9/2} x^8 + \varphi^{3/2} (\varphi (\varphi (-0.25 \varphi^3 - 1. \varphi - 1.) - 2.) + 1.) + 0.25) x^6 +$ 
 $\varphi (\varphi (0.25 \varphi^6 - 0.25 \varphi^5 + 1. \varphi^4 - 1. \varphi^3 - 2. \varphi - 1.25) + 0.25) x^5 +$ 
 $\sqrt{\varphi} (\varphi (0.25 \varphi^7 + 0.25 \varphi^6 + 0.25 \varphi^5 + 1. \varphi^4 + 2. \varphi^3 + 0.75 \varphi - 1.25) - 0.25) x^4 +$ 
 $(\varphi (\varphi (\varphi (\varphi (-0.25 \varphi (1. \varphi - 1.) (1. \varphi^2 + 1.) - 1.) + 2.) + 1.25) + 0.75) + 0.25) - 0.25) x^3 +$ 
 $\sqrt{\varphi} (\varphi (\varphi (-0.25 \varphi^5 - 0.25 \varphi^4 - 0.25 \varphi^3 - 1. \varphi - 1.) + 1.25) + 0.25) + 0.25) x^2 +$ 
 $\varphi (0.25 \varphi^6 - 0.25 \varphi^5 - 1. \varphi - 0.25) + 0.25) x)$ 

In[*]:= (* Collect and compare them *)
((1 + σ x2) (1 - τ x2))2 /. s1Rep;
FullSimplify[% == %];
Expand[%];
Collect[%, x]
Out[*]=  $x^8 + \left(-2\varphi - \frac{2}{\varphi}\right)x^6 + \left(\varphi^2 + \frac{1}{\varphi^2} + 4\right)x^4 + \left(-2\varphi - \frac{2}{\varphi}\right)x^2 + 1$ 

In[*]:= (* The palindrome of coefficients in the characteristic
matrix of U *)
{1, 0, 2 (σ - τ), 0, 7, 0, 2 (σ - τ), 0, 1};
cU = 1 + 0 x + 2 (σ - τ) x2 + 0 x3 + 7 x4 + 0 x5 + 2 (σ - τ) x6 + 0 x7 + x8 /. s1Rep;
FullSimplify[% == %%, Assumptions -> φAssumptions]
Out[*]= φ2 x = (φ + 1) x

In[*]:= (* Verify the simplification is True *)
Chop@N[% /. φRep]
Out[*]= True
```

FIG. 7. The trace, determinant, Eigenvalues, Eigenvector matrix, and characteristic polynomial coefficients of U

```
octSimplify /@FullSimplify[Eigenvalues@U, Assumptions -> φAssumptions]
Total@N[% /. φRep]
FullSimplify[Eigenvectors@U, Assumptions -> φAssumptions]
{e1+√φ, e1-√φ, e-1+√φ, e-1-√φ, e1+√φ, e1-√φ, e-1+√φ, e-1-√φ}
4.0037 + 0. i

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

CF1 = Cos[ $\frac{1}{\sqrt{\varphi}}$ ] + Cos[ $\sqrt{\varphi}$ ];
CF2 = Cos[ $\frac{1}{\sqrt{\varphi}}$ ] Cos[ $\sqrt{\varphi}$ ];
CF3 = Cos[ $\frac{1}{\sqrt{\varphi}}$ ]2 Cos[ $\sqrt{\varphi}$ ] - Cos[ $\frac{1}{\sqrt{\varphi}}$ ] Cos[ $\sqrt{\varphi}$ ]2;
CF4 = Cos[ $\frac{1}{\sqrt{\varphi}}$ ]2 + Cos[ $\sqrt{\varphi}$ ]2;

(* The palindrome of coefficients in the characteristic matrix of eIU *)
{1, -4 CF1, 4 (1 + 4 CF2 + CF4), -4 (3 CF1 + 4 CF3), 2 (3 + 4 (CF4 + 2 CF2 (CF2 + 2))), -4 (3 CF1 + 4 CF3), 4 (1 + 4 CF2 + CF4), -4 CF1, 1};
ceIU = 1 - 4 CF1 x + 4 (1 + 4 CF2 + CF4) x2 - 4 (3 CF1 + 4 CF3) x3 + 2 (3 + 4 (CF4 + 2 CF2 (CF2 + 2))) x4 - 4 (3 CF1 + 4 CF3) x5 + 4 (1 + 4 CF2 + CF4) x6 - 4 CF1 x7 + x8 /. s1Rep;
N[% /. φRep]
x8 - 4.0037 x7 + 9.67125 x6 - 15.3419 x5 + 18.0346 x4 - 15.3419 x3 + 9.67125 x2 - 4.0037 x + 1.

(* Re@eIU *)
FullSimplify[Re@eIU, Assumptions -> φAssumptions] // MatrixForm
MatrixForm

$$\begin{pmatrix} \cos\left(\frac{1}{\sqrt{\varphi}}\right) \cos\left(\frac{\varphi^2}{2}\right) & 0 & 0 & 0 & 0 & 0 & 0 & -\sin\left(\frac{1}{\sqrt{\varphi}}\right) \sin\left(\frac{\varphi^2}{2}\right) \\ 0 & \frac{1}{2} \left(\cos\left(\frac{1}{\sqrt{\varphi}}\right) + \cos(\sqrt{\varphi})\right) & 0 & 0 & 0 & 0 & \frac{1}{2} \left(\cos(\sqrt{\varphi}) - \cos\left(\frac{1}{\sqrt{\varphi}}\right)\right) & 0 \\ 0 & 0 & \frac{1}{2} \left(\cos\left(\frac{1}{\sqrt{\varphi}}\right) + \cos(\sqrt{\varphi})\right) & 0 & 0 & 0 & \frac{1}{2} \left(\cos(\sqrt{\varphi}) - \cos\left(\frac{1}{\sqrt{\varphi}}\right)\right) & 0 \\ 0 & 0 & 0 & \frac{1}{2} \left(\cos\left(\frac{1}{\sqrt{\varphi}}\right) + \cos(\sqrt{\varphi})\right) & \frac{1}{2} \left(\cos(\sqrt{\varphi}) - \cos\left(\frac{1}{\sqrt{\varphi}}\right)\right) & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} \left(\cos(\sqrt{\varphi}) - \cos\left(\frac{1}{\sqrt{\varphi}}\right)\right) & \frac{1}{2} \left(\cos\left(\frac{1}{\sqrt{\varphi}}\right) + \cos(\sqrt{\varphi})\right) & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} \left(\cos(\sqrt{\varphi}) - \cos\left(\frac{1}{\sqrt{\varphi}}\right)\right) & 0 & 0 & \frac{1}{2} \left(\cos\left(\frac{1}{\sqrt{\varphi}}\right) + \cos(\sqrt{\varphi})\right) & 0 \\ 0 & \frac{1}{2} \left(\cos(\sqrt{\varphi}) - \cos\left(\frac{1}{\sqrt{\varphi}}\right)\right) & 0 & 0 & 0 & 0 & \frac{1}{2} \left(\cos\left(\frac{1}{\sqrt{\varphi}}\right) + \cos(\sqrt{\varphi})\right) & 0 \\ -\sin\left(\frac{1}{\sqrt{\varphi}}\right) \sin\left(\frac{\varphi^2}{2}\right) & 0 & 0 & 0 & 0 & 0 & 0 & \cos\left(\frac{1}{\sqrt{\varphi}}\right) \cos\left(\frac{\varphi^2}{2}\right) \end{pmatrix}$$

Tr@%
N[% /. φRep]
2 Cos[ $\frac{1}{2\sqrt{\varphi}}$ ] Cos[ $\frac{\varphi^2}{2}$ ] + 2 Cos[ $\frac{1}{\sqrt{\varphi}}$ ] Cos[ $\sqrt{\varphi}$ ]
4.0037

(* Im@eIU *)
FullSimplify[Im@eIU, Assumptions -> φAssumptions] // MatrixForm
MatrixForm

$$\begin{pmatrix} \sin\left(\frac{1}{\sqrt{\varphi}}\right) \cos\left(\frac{\varphi^2}{2}\right) & 0 & 0 & 0 & 0 & 0 & 0 & \sin\left(\frac{\varphi^2}{2}\right) \cos\left(\frac{1}{\sqrt{\varphi}}\right) \\ 0 & -\frac{1}{2} \sin\left(\frac{1}{\sqrt{\varphi}}\right) \frac{\sin(\sqrt{\varphi})}{2} & 0 & 0 & \frac{\sin(\sqrt{\varphi})}{2} & \frac{1}{2} \sin\left(\frac{1}{\sqrt{\varphi}}\right) & 0 & 0 \\ 0 & \frac{\sin(\sqrt{\varphi})}{2} & 0 & -\frac{1}{2} \sin\left(\frac{1}{\sqrt{\varphi}}\right) \frac{\sin(\sqrt{\varphi})}{2} & \frac{\sin(\sqrt{\varphi})}{2} & \frac{1}{2} \sin\left(\frac{1}{\sqrt{\varphi}}\right) & 0 & 0 \\ 0 & 0 & -\frac{1}{2} \sin\left(\frac{1}{\sqrt{\varphi}}\right) \frac{\sin(\sqrt{\varphi})}{2} & \frac{\sin(\sqrt{\varphi})}{2} & \frac{\sin(\sqrt{\varphi})}{2} & \frac{1}{2} \sin\left(\frac{1}{\sqrt{\varphi}}\right) & 0 & 0 \\ 0 & 0 & \frac{1}{2} \sin\left(\frac{1}{\sqrt{\varphi}}\right) \frac{\sin(\sqrt{\varphi})}{2} & -\frac{\sin(\sqrt{\varphi})}{2} & \frac{\sin(\sqrt{\varphi})}{2} & \frac{1}{2} \sin\left(\frac{1}{\sqrt{\varphi}}\right) & 0 & 0 \\ 0 & \frac{\sin(\sqrt{\varphi})}{2} & 0 & \frac{1}{2} \sin\left(\frac{1}{\sqrt{\varphi}}\right) \frac{\sin(\sqrt{\varphi})}{2} & -\frac{\sin(\sqrt{\varphi})}{2} & \frac{1}{2} \sin\left(\frac{1}{\sqrt{\varphi}}\right) & 0 & 0 \\ 0 & \frac{1}{2} \sin\left(\frac{1}{\sqrt{\varphi}}\right) \frac{\sin(\sqrt{\varphi})}{2} & 0 & 0 & 0 & \frac{\sin(\sqrt{\varphi})}{2} & -\frac{1}{2} \sin\left(\frac{1}{\sqrt{\varphi}}\right) & 0 \\ \sin\left(\frac{\varphi^2}{2}\right) \cos\left(\frac{1}{\sqrt{\varphi}}\right) & 0 & 0 & 0 & 0 & 0 & 0 & \sin\left(\frac{1}{\sqrt{\varphi}}\right) \cos\left(\frac{\varphi^2}{2}\right) \end{pmatrix}$$

Tr@%
Chop@N[% /. φRep]
-2 Sin[ $\frac{1}{2\sqrt{\varphi}}$ ] Cos[ $\frac{\varphi^2}{2}$ ] - Sin[ $\frac{1}{\sqrt{\varphi}}$ ] + Sin[ $\sqrt{\varphi}$ ]
0

(* Re@eIU *)
FullSimplify[Re@eIU /. φRep, Assumptions -> φAssumptions] // MatrixForm
MatrixForm

$$\begin{pmatrix} 0.500463 & 0. & 0. & 0. & 0. & 0. & -0.206111 & 0. \\ 0. & 0.500463 & 0. & 0. & 0. & -0.206111 & 0. & 0. \\ 0. & 0. & 0.500463 & -0.206111 & 0. & 0. & 0. & 0. \\ 0. & 0. & -0.206111 & 0.500463 & 0. & 0. & 0. & 0. \\ 0. & -0.206111 & 0. & 0. & 0.500463 & 0. & 0. & 0. \\ 0. & 0. & -0.206111 & 0. & 0. & 0.500463 & 0. & 0. \\ -0.206111 & 0. & 0. & 0. & 0. & 0. & 0.500463 & 0. \end{pmatrix}$$


(* Im@eIU *)
FullSimplify[Im@eIU /. φRep, Assumptions -> φAssumptions] // MatrixForm
MatrixForm

$$\begin{pmatrix} -0.124029 & 0. & 0. & 0. & 0. & 0. & -0.831668 & 0. \\ 0. & -0.35382 & 0.477849 & 0. & 0. & 0.477849 & 0.35382 & 0. \\ 0.477849 & 0. & -0.35382 & 0.35382 & 0. & 0.477849 & 0. & 0. \\ 0. & -0.35382 & 0.477849 & 0.477849 & 0.35382 & 0. & 0. & 0. \\ 0. & 0.35382 & 0.477849 & 0.477849 & -0.35382 & 0. & 0. & 0. \\ 0.477849 & 0. & 0.35382 & -0.35382 & 0. & 0.477849 & 0. & 0. \\ 0.35382 & 0.477849 & 0. & 0. & 0.477849 & -0.35382 & 0. & 0. \\ -0.831668 & 0. & 0. & 0. & 0. & 0. & -0.124029 & 0. \end{pmatrix}$$

```

FIG. 8. The Eigenvalues, Eigenvector matrix, and characteristic polynomial coefficients of the unitary form of U as e^{iU} showing a $\text{Tr@Re@}e^{iU} \approx 4$ and a traceless imaginary part

III. QUATERNIONIC WEYL ORBIT CONSTRUCTION

The content within this paper was generated using a computational environment the author has written in *Mathematica*TM by *Wolfram Research, Inc.*. In order to deal effectively with quaternions, it supplants the native Quaternion package with a more flexible symbolic octonion (\mathbb{O}) capability. This allows for the selection of a multiplication table from any of the 480 possible octonion tables, including their split and bi-octonion forms. It also handles the sedenion forms as well and has been used to verify the octonion forms of E_8 from Koca[1], Dixon[15], Pushpa and Bisht[16], R. A. Wilson, Dray, and Monague[17], including the complexified octonions of Günaydin-Gürsey[18] and Furey[19]. To ensure that our quaternion (and bi-quaternion) math is consistent with the standard multiplication convention related to quaternions, we need to select one of the 48 octonions with a first triad of 123 and a Cayley-Dickson construction where e_4 - e_7 quadrant multiplication remains within the quadrant. See Fig. 9 showing the selected triads, Fano plane, and multiplication table of the octonion used in this and several of the referenced papers¹.

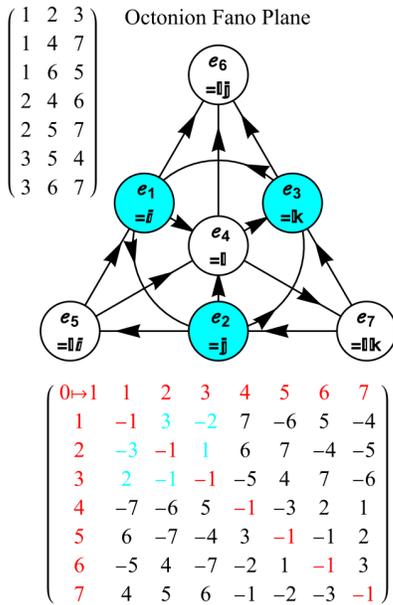


FIG. 9. The selected octonion Fano plane mnemonic and multiplication table based on its 7 structure constant triads. The first triad (123) defines standard convention for quaternions.

¹ It is interesting to note that this particular octonion is close to (but not) palindromic. Using an algorithmic identification and construction of all of the possible 480 unique permutations of octonions[20], we find that a small change in triads to {123,145,167,264,257,347,356} with 5↔7 ordering swaps creates a palindromic E_8 . This octonion is shown in Fig. 10.

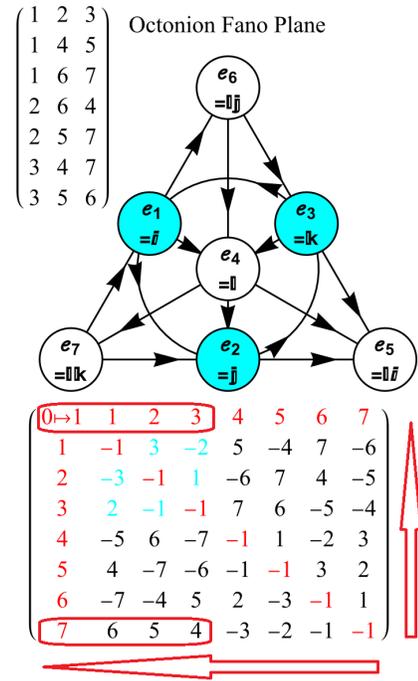


FIG. 10. An alternative set of structure constant triads, octonion Fano plane mnemonic, and multiplication table, with decorations showing the palindromic multiplication.

It has been shown that the 3D symmetry groups of A_3 , B_3 , and H_3 [3] and 4D symmetry groups of A_4 , D_4 , F_4 , and H_4 are related to the higher dimensional groups of D_6 and E_8 [5][9]. A quaternionic Weyl group orbit $O(\Lambda)=W(H_4)=I$ of order 120 can be constructed from H_3 which generates some of the Platonic, Archimedean and dual Catalan solids shown in Appendix B Fig. 18, including their irregular and chiral forms[4]. The polytopes for a particular orbit of $O(\Lambda)=W(\text{group})$ are generated using a function $\Lambda[\text{group}_-, \text{orbit}_-, \text{perm}_- : \text{"Rotate"}]$, where perm can be one of 18 combinations of sign and position permutation functions (e.g. "oSign" gives all odd sign permutations and cyclic rotations of position and the default "Rotate" gives all sign permutations of cyclically rotated positions). The first column in these figures show the set of calls to the Λ function. This same method is used to generate the H_4 -based 4-polytopes of the 120-cell and 600-cell shown in Appendix A Figs. 14-16.

The A_3 in A_4 group embedding of $SU(5) \supset SU(4) \otimes U_1$ [5] are shown in Appendix C Fig. 20 in combination with these 3 and 4-polytope visualizations.²

We identify the rectified parent orbit (0100) of $W(D_4)$ as the self-dual 24-cell T, which is the combination of the 4D octahedron (aka. 16-cell) and the 4D cube (aka. 8-cell

² In the methods and coding descriptions, since Mamone[6] identifies the 5-cell as S, but Koca uses S to identify the (S) nub 24-cell (a convention which we use here), Mamone's A_4 -based 5-cell is now identified as A which is the 4D version of the tetrahedron.

```
(* Show T vertices *)
checkVertices[T, False, True, True, False, False, False]
Out[*]= List length= 24 and it is symbolic octonion
```

$$\text{Math} = \begin{pmatrix} 1 & \frac{1}{2}(-1 - e_1 - e_2 - e_3) \\ 2 & \frac{1}{2}(-1 - e_1 - e_2 + e_3) \\ 3 & \frac{1}{2}(-1 - e_1 + e_2 - e_3) \\ 4 & \frac{1}{2}(-1 - e_1 + e_2 + e_3) \\ 5 & \frac{1}{2}(-1 + e_1 - e_2 - e_3) \\ 6 & \frac{1}{2}(-1 + e_1 - e_2 + e_3) \\ 7 & \frac{1}{2}(-1 + e_1 + e_2 - e_3) \\ 8 & \frac{1}{2}(-1 + e_1 + e_2 + e_3) \\ 9 & \frac{1}{2}(1 - e_1 - e_2 - e_3) \\ 10 & \frac{1}{2}(1 - e_1 - e_2 + e_3) \\ 11 & \frac{1}{2}(1 - e_1 + e_2 - e_3) \\ 12 & \frac{1}{2}(1 - e_1 + e_2 + e_3) \\ 13 & \frac{1}{2}(1 + e_1 - e_2 - e_3) \\ 14 & \frac{1}{2}(1 + e_1 - e_2 + e_3) \\ 15 & \frac{1}{2}(1 + e_1 + e_2 - e_3) \\ 16 & \frac{1}{2}(1 + e_1 + e_2 + e_3) \\ 17 & -e_3 \\ 18 & -e_2 \\ 19 & -e_1 \\ 20 & -1 \\ 21 & e_3 \\ 22 & e_2 \\ 23 & e_1 \\ 24 & 1 \end{pmatrix}$$

```
Numeric =
```

$$\begin{pmatrix} 1 & -0.5 - 0.5 e_1 - 0.5 e_2 - 0.5 e_3 \\ 2 & -0.5 - 0.5 e_1 - 0.5 e_2 + 0.5 e_3 \\ 3 & -0.5 - 0.5 e_1 + 0.5 e_2 - 0.5 e_3 \\ 4 & -0.5 - 0.5 e_1 + 0.5 e_2 + 0.5 e_3 \\ 5 & -0.5 + 0.5 e_1 - 0.5 e_2 - 0.5 e_3 \\ 6 & -0.5 + 0.5 e_1 - 0.5 e_2 + 0.5 e_3 \\ 7 & -0.5 + 0.5 e_1 + 0.5 e_2 - 0.5 e_3 \\ 8 & -0.5 + 0.5 e_1 + 0.5 e_2 + 0.5 e_3 \\ 9 & 0.5 - 0.5 e_1 - 0.5 e_2 - 0.5 e_3 \\ 10 & 0.5 - 0.5 e_1 - 0.5 e_2 + 0.5 e_3 \\ 11 & 0.5 - 0.5 e_1 + 0.5 e_2 - 0.5 e_3 \\ 12 & 0.5 - 0.5 e_1 + 0.5 e_2 + 0.5 e_3 \\ 13 & 0.5 + 0.5 e_1 - 0.5 e_2 - 0.5 e_3 \\ 14 & 0.5 + 0.5 e_1 - 0.5 e_2 + 0.5 e_3 \\ 15 & 0.5 + 0.5 e_1 + 0.5 e_2 - 0.5 e_3 \\ 16 & 0.5 + 0.5 e_1 + 0.5 e_2 + 0.5 e_3 \\ 17 & 0. - 1. e_3 \\ 18 & 0. - 1. e_2 \\ 19 & 0. - 1. e_1 \\ 20 & -1. \\ 21 & 0. + 1. e_3 \\ 22 & 0. + 1. e_2 \\ 23 & 0. + 1. e_1 \\ 24 & 1. \end{pmatrix}$$

```
(* Show T' vertices *)
checkVertices[Tp, False, True, True, False, False, False]
Out[*]= List length= 24 and it is symbolic octonion
```

$$\text{Math} = \begin{pmatrix} 1 & \frac{1+e_1}{\sqrt{2}} \\ 2 & \frac{1+e_2}{\sqrt{2}} \\ 3 & \frac{1+e_3}{\sqrt{2}} \\ 4 & \frac{-1+e_3}{\sqrt{2}} \\ 5 & \frac{-1+e_2}{\sqrt{2}} \\ 6 & \frac{-1+e_1}{\sqrt{2}} \\ 7 & \frac{e_1+e_2}{\sqrt{2}} \\ 8 & \frac{e_1+e_3}{\sqrt{2}} \\ 9 & \frac{e_1+e_2}{\sqrt{2}} \\ 10 & \frac{e_1+e_3}{\sqrt{2}} \\ 11 & \frac{e_2+e_3}{\sqrt{2}} \\ 12 & \frac{e_2+e_3}{\sqrt{2}} \\ 13 & \frac{e_2-e_3}{\sqrt{2}} \\ 14 & \frac{e_2+e_3}{\sqrt{2}} \\ 15 & \frac{e_1-e_2}{\sqrt{2}} \\ 16 & \frac{e_1-e_3}{\sqrt{2}} \\ 17 & \frac{e_1+e_3}{\sqrt{2}} \\ 18 & \frac{e_1+e_2}{\sqrt{2}} \\ 19 & \frac{-1+e_1}{\sqrt{2}} \\ 20 & \frac{-1+e_2}{\sqrt{2}} \\ 21 & \frac{-1+e_3}{\sqrt{2}} \\ 22 & \frac{1+e_3}{\sqrt{2}} \\ 23 & \frac{1+e_2}{\sqrt{2}} \\ 24 & \frac{1+e_1}{\sqrt{2}} \end{pmatrix}$$

```
Numeric =
```

$$\begin{pmatrix} 1 & -0.70711 - 0.70711 e_1 \\ 2 & -0.70711 - 0.70711 e_2 \\ 3 & -0.70711 - 0.70711 e_3 \\ 4 & -0.70711 + 0.70711 e_3 \\ 5 & -0.70711 + 0.70711 e_2 \\ 6 & -0.70711 + 0.70711 e_1 \\ 7 & 0. - 0.70711 e_1 - 0.70711 e_2 \\ 8 & 0. - 0.70711 e_1 - 0.70711 e_3 \\ 9 & 0. - 0.70711 e_1 + 0.70711 e_3 \\ 10 & 0. - 0.70711 e_1 + 0.70711 e_2 \\ 11 & 0. - 0.70711 e_2 - 0.70711 e_3 \\ 12 & 0. - 0.70711 e_2 + 0.70711 e_3 \\ 13 & 0. + 0.70711 e_2 - 0.70711 e_3 \\ 14 & 0. + 0.70711 e_2 + 0.70711 e_3 \\ 15 & 0. + 0.70711 e_1 - 0.70711 e_2 \\ 16 & 0. + 0.70711 e_1 - 0.70711 e_3 \\ 17 & 0. + 0.70711 e_1 + 0.70711 e_3 \\ 18 & 0. + 0.70711 e_1 + 0.70711 e_2 \\ 19 & 0.70711 - 0.70711 e_1 \\ 20 & 0.70711 - 0.70711 e_2 \\ 21 & 0.70711 - 0.70711 e_3 \\ 22 & 0.70711 + 0.70711 e_1 \\ 23 & 0.70711 + 0.70711 e_2 \\ 24 & 0.70711 + 0.70711 e_3 \end{pmatrix}$$

FIG. 11. The values of the D_4 24-cell T and its alternate T'

```
In[*]= (* Generate the A' parent Weyl orbit *)
AA4[{0, 1, 4, 2, 3}, {1, 0, 0, 0}]
```

$$\text{Out[*]} = \begin{pmatrix} -\frac{1}{\sqrt{2}} & 0 & \frac{\varphi}{\sqrt{10}} & \frac{1}{\sqrt{10}\varphi} \\ \frac{1}{\sqrt{2}} & 0 & \sqrt{\frac{2}{5}}\varphi - \frac{\varphi}{\sqrt{10}} & \frac{\sqrt{\frac{2}{5}}}{\varphi} - \frac{1}{\sqrt{10}\varphi} \\ 0 & -\frac{1}{\sqrt{2}} & \frac{\varphi^2}{\sqrt{10}} - \sqrt{\frac{2}{5}}\varphi & -\frac{1}{\sqrt{10}\varphi^2} - \frac{\sqrt{\frac{2}{5}}}{\varphi} \\ 0 & \frac{1}{\sqrt{2}} & \sqrt{\frac{2}{5}} - \frac{\varphi^2}{\sqrt{10}} & \frac{1}{\sqrt{10}\varphi^2} + \frac{\sqrt{\frac{2}{5}}}{\varphi} \\ 0 & 0 & -\sqrt{\frac{2}{5}} & \sqrt{\frac{2}{5}} \end{pmatrix} \text{NoneNoSign}$$

```
In[*]= (* Scale by -\frac{\sqrt{5}}{2} to match conventional A scaling *)
Aplist = oct2List@biQuaternion[-\frac{\sqrt{5}}{2} #] & /@ #[[1, 1]]
```

$$\text{Out[*]} = \begin{pmatrix} \frac{\sqrt{5}}{2} & 0 & -\frac{\varphi}{2\sqrt{2}} & \frac{1}{2\sqrt{2}\varphi} & 0 & 0 & 0 & 0 \\ -\frac{\sqrt{5}}{2} & 0 & -\frac{\varphi}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}\varphi} & 0 & 0 & 0 & 0 \\ 0 & \frac{\sqrt{5}}{2} & \frac{(\varphi-2)\varphi}{2\sqrt{2}} & \frac{2\varphi+1}{2\sqrt{2}\varphi^2} & 0 & 0 & 0 & 0 \\ 0 & -\frac{\sqrt{5}}{2} & \frac{(\varphi^2-2)}{2\sqrt{2}} & \frac{-\varphi^2-2}{2\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \end{pmatrix}$$

```
In[*]= (* Put it into symbolic octonion form *)
Ap = octSimplify@octonion /@ #;
% // MatrixForm
```

$$\text{Out[*]} // \text{MatrixForm} = \begin{pmatrix} -\frac{\varphi e_2}{2\sqrt{2}} - \frac{e_1}{2\sqrt{2}\varphi} + \frac{\sqrt{5}}{2} \\ -\frac{\varphi e_3}{2\sqrt{2}} - \frac{e_1}{2\sqrt{2}\varphi} - \frac{\sqrt{5}}{2} \\ \frac{(2e_1+1)e_3}{2\sqrt{2}\varphi^2} - \frac{(\varphi-2)\varphi e_2}{2\sqrt{2}} + \frac{1}{2}\sqrt{\frac{5}{2}} e_1 \\ \frac{(\varphi^2-2)e_2}{2\sqrt{2}} - \frac{(\frac{1}{2}-2)e_3}{2\sqrt{2}} - \frac{1}{2}\sqrt{\frac{5}{2}} e_1 \\ \frac{e_2}{\sqrt{2}} - \frac{e_3}{\sqrt{2}} \end{pmatrix}$$

```
In[*]= (* Display vertex value *)
checkVertices[%, False, True, True, False, False, False]
```

$$\text{Out[*]} = \text{List length= 5 and it is symbolic octonion}$$

$$\text{Math} = \begin{pmatrix} 1 & \frac{1}{4}(1+\sqrt{5})e_2 + \frac{1}{2}\sqrt{5}(1+\sqrt{5}) \\ 2 & \frac{1}{4}(1+\sqrt{5})e_1 + \frac{1}{2}\sqrt{5}(1+\sqrt{5}) \\ 3 & \frac{\sqrt{3}}{4}(1+\sqrt{5})e_1 + \frac{\sqrt{3}}{4}(1+\sqrt{5})e_2 + \frac{1}{2}\sqrt{5}(1+\sqrt{5}) \\ 4 & \frac{\sqrt{2}(1+\sqrt{5})e_1(-2+2\sqrt{5})e_2 + (1+\sqrt{5})e_3}{(1+\sqrt{5})^2} \\ 5 & \frac{\sqrt{2}(1+\sqrt{5})e_1(-2+2\sqrt{5})e_2 + (1+\sqrt{5})e_3}{(1+\sqrt{5})^2} \end{pmatrix}$$

```
Numeric =
```

$$\begin{pmatrix} 1 & -0.57206 e_2 - 0.21851 e_3 + 0.79057 \\ 2 & -0.57206 e_2 - 0.21851 e_3 - 0.79057 \\ 3 & 0.79057 e_1 + 0.21851 e_2 + 0.57206 e_3 + 0. \\ 4 & -0.79057 e_1 + 0.21851 e_2 + 0.57206 e_3 + 0. \\ 5 & 0.70711 e_2 - 0.70711 e_3 + 0. \end{pmatrix}$$

```
In[*]= (* Simplify quaternion multiplication using prqr which also handles lists,
We scale up/down by 4 for symbolic clarity.
Please note the double struck A to avoid stepping on LieArt *)
A = \frac{1}{4} octonion@biQuaternion[# /. @Rep] * /. slRep & /@
(* @Rep replaces the symbolic forms \varphi \to (\sqrt{5}+1)/2, also note the conjugation
(4 oct2Quat@# & /@ Flatten@prq[cp, 1, Ap]);
```

```
In[*]= checkVertices[A /. e_0 -> 1, False, True, True, False, False, False]
```

$$\text{Out[*]} = \text{List length= 5 and it is symbolic octonion}$$

$$\text{Math} = \begin{pmatrix} 1 & \frac{1}{4}(\sqrt{5} e_1 - \sqrt{5} e_2 + \sqrt{5} e_3 + 1) \\ 2 & \frac{1}{4}(\sqrt{5} e_1 + \sqrt{5} e_2 - \sqrt{5} e_3 + 1) \\ 3 & \frac{1}{4}(-\sqrt{5} e_1 + \sqrt{5} e_2 + \sqrt{5} e_3 + 1) \\ 4 & \frac{1}{4}(-\sqrt{5} e_1 - \sqrt{5} e_2 - \sqrt{5} e_3 + 1) \\ 5 & -1 \end{pmatrix}$$

```
Numeric =
```

$$\begin{pmatrix} 1 & 0.55902 e_1 - 0.55902 e_2 + 0.55902 e_3 + 0.25 \\ 2 & 0.55902 e_1 + 0.55902 e_2 - 0.55902 e_3 + 0.25 \\ 3 & -0.55902 e_1 + 0.55902 e_2 + 0.55902 e_3 + 0.25 \\ 4 & -0.55902 e_1 - 0.55902 e_2 - 0.55902 e_3 + 0.25 \\ 5 & -1. \end{pmatrix}$$

FIG. 12. Explicit *Mathematica*TM computation of A from the AA4[Λ, orbit_] generated A'

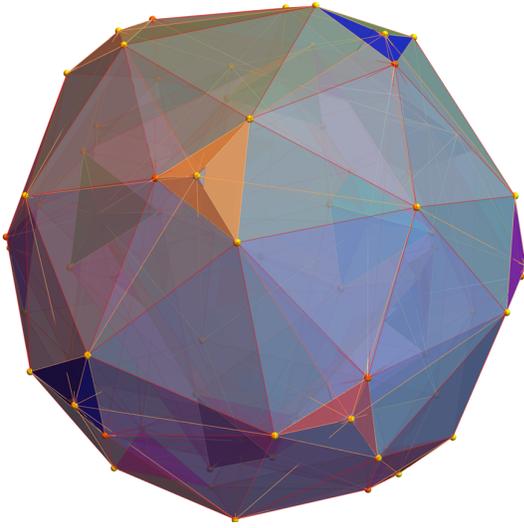


FIG. 13. Visualization of the 144 root vertices of $S'+T+T'$ now identified as the dual snub 24-cell

with a 3D hull of the cuboctahedron derived from the tri-rectified (0001) $W(BC_4)$. Due to the $W(D_4)$ Coxeter-Dynkin diagram triality symmetry, T' is identified with any of 3 end nodes as parent and others as bi-rectified and tri-rectified orbits $\{(1000), (0010), (0001)\}$ each with 8 vertices of 2-component (vector) quaternions and has a 3D hull of the rhombic dodecahedron. See Fig. 11 for their specific symbolic and numeric values. Of course, it has also been shown that the root system of $F_4 = T \oplus T'$.

From T (and T') we can take any one vertex to define a c (and $c'=cp$) respectively. For this paper, we use as an example $c=t_1$ from eq. (18) from Koca[3] T (and T') shown as #13 in Fig. 11 such that $c=\frac{1}{2}(1+e_1-e_2-e_3)$ (and $c'=\frac{e_2-e_3}{\sqrt{2}}$). Here c' is used with A' to generate the parent $W(A_4)$, or simply A as the 5-cell[3]. Specifically, $A=(c' \circ A')^*$ with $A'=\text{AA4}[\{0, 1, 4, 2, 3\}, \{1, 0, 0, 0\}]$.³ See Fig. 12 for the explicit *Mathematica*TM computation related to A and A' .

The snub orbit (0000) of $W(D_4)$ will generate the vertices of the snub 24-cell or $S=I-T$, as with the alternate

snub 24-cell $S'=I'-T'$ as shown in (7) and (8). We can generate S (or S') by taking the odd (or even) sign and cyclic position permutations of a seed quaternion $p \in S$ (or S') to be assigned to α (or β) for generating S (or S') respectively. There are only 48 that satisfy the necessary constraint where a unit normed $p^5 = \pm 1$. Those quaternions that satisfy the constraint are identified with an $*$ in Appendix D. For this paper, we selected from the 96 permutations of S $\alpha = \frac{1}{2} \left(\frac{1}{\varphi} + \varphi e_2 + e_1 \right)$ (and S' for $\beta = \frac{-\varphi - e_2 + \sqrt{5}e_1}{\sqrt{8}}$). This process of generating the snub 24-cell can be visualized as generating four quaternion 4D rotations of T (and T'). The 3D hulls of T are shown in Fig. 15.

$$\begin{aligned} S &= I - T = \sum_{i=1}^4 \alpha^i \circ T \\ \text{or} \\ I &= \text{prq}[\alpha^{0-4}, 1, T] \end{aligned} \quad (7)$$

$$\begin{aligned} S' &= I' - T' = \sum_{i=1}^4 \beta^i \circ T' \\ \text{or} \\ I' &= \text{prq}[\beta^{0-4}, 1, T'] \end{aligned} \quad (8)$$

The 3D hulls for one copy of I (or φI) are represented in Fig. 14 hulls $\{2,3,5\}$ (or $\{6,7,8\}$) respectively plus $1/2$ of the vertices in hull 4. The vertex values of I are listed in either of the center columns of Appendix D Fig. 22 or Fig. 23.

Koca[3] has also identified the dual to the snub 24-cell as being made up of the 144 root vertices of $S'+T+T'$. This 4-polytope is visualized in Fig. 13.

The equations for the generation of J (and J') are shown in (9) and (10). As it was for I (and I') vertices each mapping to 5 quaternion rotations of T (and T'), J (and J') vertices each map to 5 quaternion rotations of I (and I') or 25 quaternion rotations of T (and T'). Given the isomorphism between each E_8 root vertex and 4 copies of I (i.e. L and R each at unit and φ scales) as demonstrated in Section II, this means quaternionic Weyl orbit construction, when used with U and mapLR , provides for an explicit map between each of the 240 E_8 root vertices and 10 J (or J') vertices (i.e. $10=2(L \oplus R) \times 5$ quaternion rotations of each I (or I') vertex).

$$\begin{aligned} J &= \sum_{i=0}^4 c' \circ \bar{\alpha}^{\dagger i} \circ \alpha^i \circ T \\ \text{or} \\ J &= \text{prq}[A', \alpha^{0-4}, T] \end{aligned} \quad (9)$$

$$\begin{aligned} J' &= \sum_{i=0}^4 c \circ \bar{\beta}^{\dagger i} \circ \beta^i \circ T' \\ \text{or} \\ J' &= \text{prq}[A', \beta^{0-4}, T'] \end{aligned} \quad (10)$$

See Figs. 16-17 for the 120-cell (J) and its alternate (J') as generated by $J=\text{prq}[A', 1, I]$ and $J'=\text{prq}[A', 1, I']$ respectively.

³ The 4-polytopes for a particular orbit of $O(\Lambda)=W(\text{group})$ are generated using a function $\Lambda[\text{group}, \text{orbit}, \text{perm}]$ which is called by $\text{AA4}[\Lambda, \text{orbit}]$ for the subgroup embeddings in A_4 as described in [5]. In addition, $\text{SmallCircle}(\circ)$ is the symbolic operator for quaternion (octonion) multiplication that operates across lists, along with the expected symbolic exponentials ($*$ and \dagger) for Conjugate and ConjugateTranspose respectively. The function $\text{prq}[p, r, q, \text{left:False}] := \text{If}[\text{left}, (p \circ r) \circ q, p \circ (r \circ q)]$ implements the operation of $[p, q]:r$ from eq. (6) in [3], which is defined for any combinations of inputs as elements or lists in order to add flexibility to quaternion and octonion operators, including left or right (default) non-commutative multiplication ordering. Other operators are also available for scalar product(\oplus), scalar product(\ominus), commutator(\odot), anti-commutator(\wedge), derivation(\square), Kronecker product(\otimes), and octExp for exponential powers of octonions.

IV. CONCLUSION

This paper has given an explicit isomorphic mapping from the 240 \mathbb{R}^8 root E_8 Gosset 4_{21} 8-polytope to two φ scaled copies of the 120 root H_4 600-cell quaternion 4-polytope using \mathbb{U} . It has also shown the inverse map from a single H_4 600-cell to E_8 using a $4\text{D} \leftrightarrow 8\text{D}$ chiral $L \leftrightarrow R$ mapping function, φ scaling, and \mathbb{U}^{-1} . This approach has shown that there are actually four copies of each 600-cell living within E_8 in the form of chiral $H_{4L} \oplus \varphi H_{4L} \oplus H_{4R} \oplus \varphi H_{4R}$ roots. In addition, it has demonstrated a quaternion Weyl orbit construction of H_4 -based 4-polytopes that provides an explicit map from

E_8 to four copies of the tri-rectified Coxeter-Dynkin diagram of H_4 , namely the 120-cell of order 600. Taking advantage of this property promises to open the door to as yet unexplored chiral E_8 -based Grand Unified Theories or GUTs. It is anticipated that these visualizations and connections will be useful in discovering new insights into unifying the mathematical symmetries as they relate to unification in theoretical physics.

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, E8 Lattice with Octonions and Icosians, CERN, 1211 Geneva 23, Switzerland (1989).
- [2] M. Koca and N. Koca, Quaternionic Roots of E8 Related Coxeter Graphs and Quasicrystals, *Turkish Journal of Physics* **22**, 421 (1998).
- [3] M. Koca, M. Al-Ajmi, and N. O. Koca, Quaternionic representation of snub 24-cell and its dual polytope derived from e8 root system, *Linear Algebra and its Applications* **434**, 977 (2011).
- [4] M. Koca, N. O. Koca, and M. Al-Shueili, Chiral Polyhedra Derived From Coxeter Diagrams and Quaternions, ArXiv e-prints math.ph (2011), [arXiv:1006.3149 \[math-ph\]](https://arxiv.org/abs/1006.3149).
- [5] M. Koca, N. O. Koca, and M. Al-Ajmi, 4d-polytopes and their dual polytopes of the coxeter group a4 represented by quaternions, *International Journal of Geometric Methods in Modern Physics* **09**, 1250035 (2012).
- [6] S. Mamone, G. Pileio, and M. H. Levitt, Orientational sampling schemes based on four dimensional polytopes, *Symmetry* **2**, 1423 (2010).
- [7] J. H. Conway, R. H. Hardin, and N. J. A. Sloane, Packing Lines, Planes, etc.: Packings in Grassmannian Space, ArXiv e-prints math.CO (2002), [arXiv:math/0208004 \[math.CO\]](https://arxiv.org/abs/math/0208004).
- [8] D. A. Richter, Triacontagonal coordinates for the E8 root system, ArXiv e-prints math.GM (2007), [arXiv:0704.3091 \[math.GM\]](https://arxiv.org/abs/0704.3091).
- [9] P. P. Dechant, The birth of E8 out of the spinors of the icosahedron, *Proceedings of the Royal Society of London Series A* **472**, 20150504 (2016), [arXiv:1602.05985 \[math-ph\]](https://arxiv.org/abs/1602.05985).
- [10] J. C. Baez, From the Icosahedron to E8, ArXiv e-prints math.HO (2017), [arXiv:1712.06436 \[math.HO\]](https://arxiv.org/abs/1712.06436).
- [11] J. G. Moxness, The 3D Visualization of E8 using an H4 Folding Matrix, [www.vixra.org/abs/1411.0130](https://vixra.org/abs/1411.0130) (2014).
- [12] J. G. Moxness, Mapping the fourfold H4 600-cells emerging from E8, [www.vixra.org/abs/1808.0107](https://vixra.org/abs/1808.0107) (2018).
- [13] J. G. Moxness, Unimodular rotation of E8 to H4 600-cells, [www.vixra.org/abs/1910.0345](https://vixra.org/abs/1910.0345) (2019).
- [14] J. G. Moxness, 3D Polytope Hulls of E8 4-21, 2-41, and 1-42, [www.vixra.org/abs/2005.0200](https://vixra.org/abs/2005.0200) (2020).
- [15] G. Dixon, Integral octonions, octonion xy-product, and the leech lattice, ArXiv e-prints math.th (2010), [arXiv:1011.2541 \[hep-th\]](https://arxiv.org/abs/1011.2541).
- [16] Pushpa, P. S. Bisht, T. Li, and O. P. S. Negi, Quaternion octonion reformulation of grand unified theories, *International Journal of Theoretical Physics* **51**, 3228 (2012).
- [17] R. A. Wilson, T. Dray, and C. A. Manogue, An octonionic construction of e8 and the lie algebra magic square, *Innovations in Incidence Geometry: Algebraic, Topological and Combinatorial* **20**, 611 (2023).
- [18] M. Günaydin and F. Gürsey, Quark structure and octonions, *Journal of Mathematical Physics* **14**, 1651 (2003), https://pubs.aip.org/aip/jmp/article-pdf/14/11/1651/8805448/1651_1_online.pdf.
- [19] C. Furey, $Su(3)_c \times su(2)_L \times u(1)_Y (\times u(1)_X)$ as a symmetry of division algebraic ladder operators, *The European Physical Journal C* **78**, 10.1140/epjc/s10052-018-5844-7 (2018).
- [20] J. G. Moxness, The Comprehensive Split Octonions and their Fano Planes, vixra.org/abs/1503.0228 (2013).
- [21] R. M. Fonseca, GroupMath: A mathematica package for group theory calculations, *Computer Physics Communications* **267**, 108085 (2021).
- [22] P. Grozman and D. Leites, Lie superalgebra structures, *Czechoslovak Journal of Physics* **54**, 1313 (2004).

Appendix A: Concentric hulls from Platonic 3D projection with numeric and symbolic norm distances
Figs. 14-17

Appendix B: Archimedean and dual Catalan solids
Fig. 18

Appendix C: Maximal $SO(16)=D_8$ related embeddings of E_8 at height 248
Figs. 19-20

Appendix D: MathematicaTM code and output showing $E_8 \leftrightarrow H_4$ isomorphism
Figs. 21-23

ListName= C60atE8

Dims used={1, 2, 3}
 tallyList={4, 24, 40, 48}
 {30, 40, 24, 30}

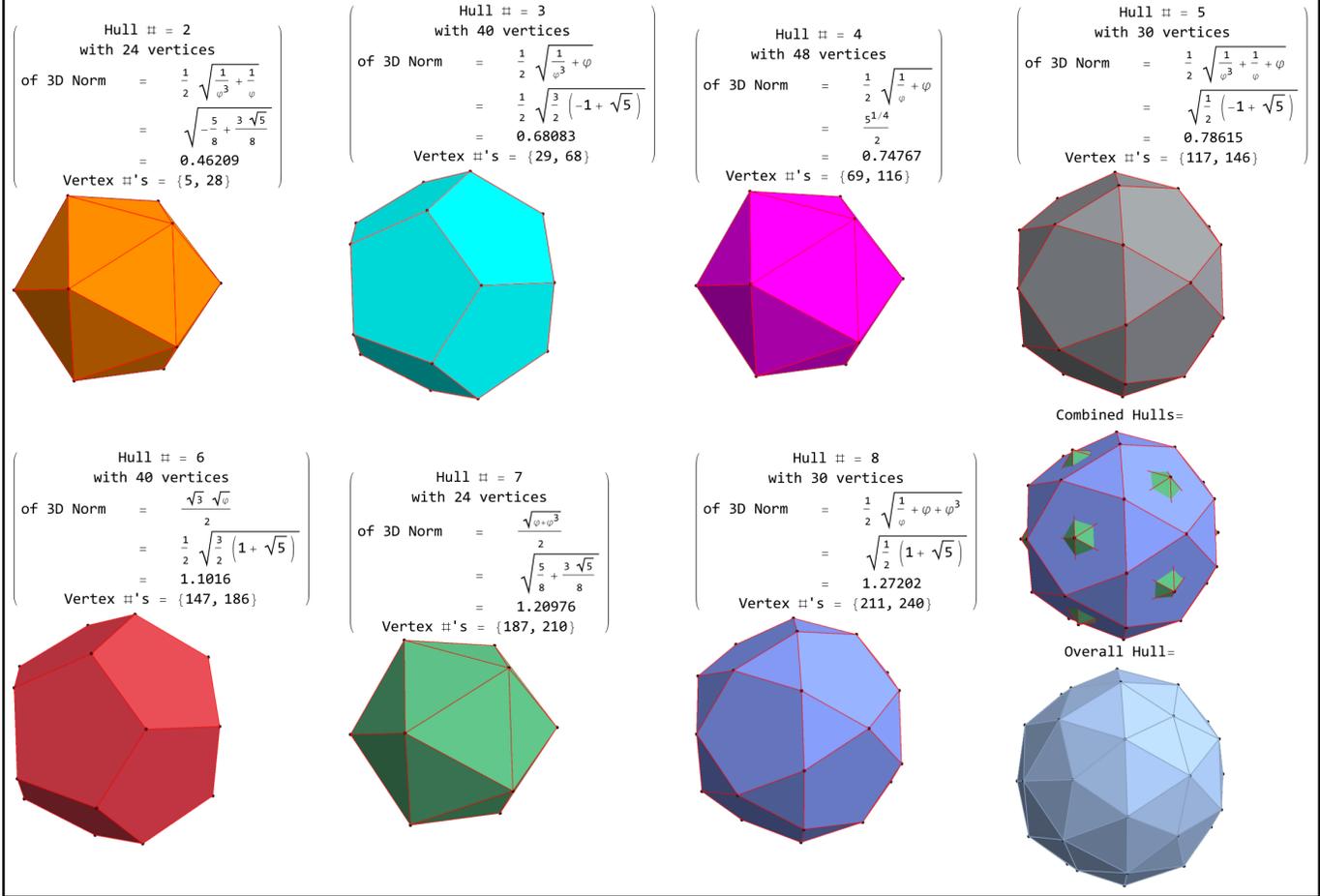


FIG. 14. Concentric hulls of 4_{21} in Platonic 3D projection with numeric and symbolic norm distances

```

In[ ]:=
Ip = Flatten@prq[octExpa, 1, Tp];
IpRnd = rndOct /@%;
IpList = oct2List@# & /@%%;
hulls3DPerms["IpList", False, , 1]
ListName= IpList

```

Dims used={1, 2, 3}
 tallyList={8, 8, 12, 24}
 {24, 24, 8, 12}

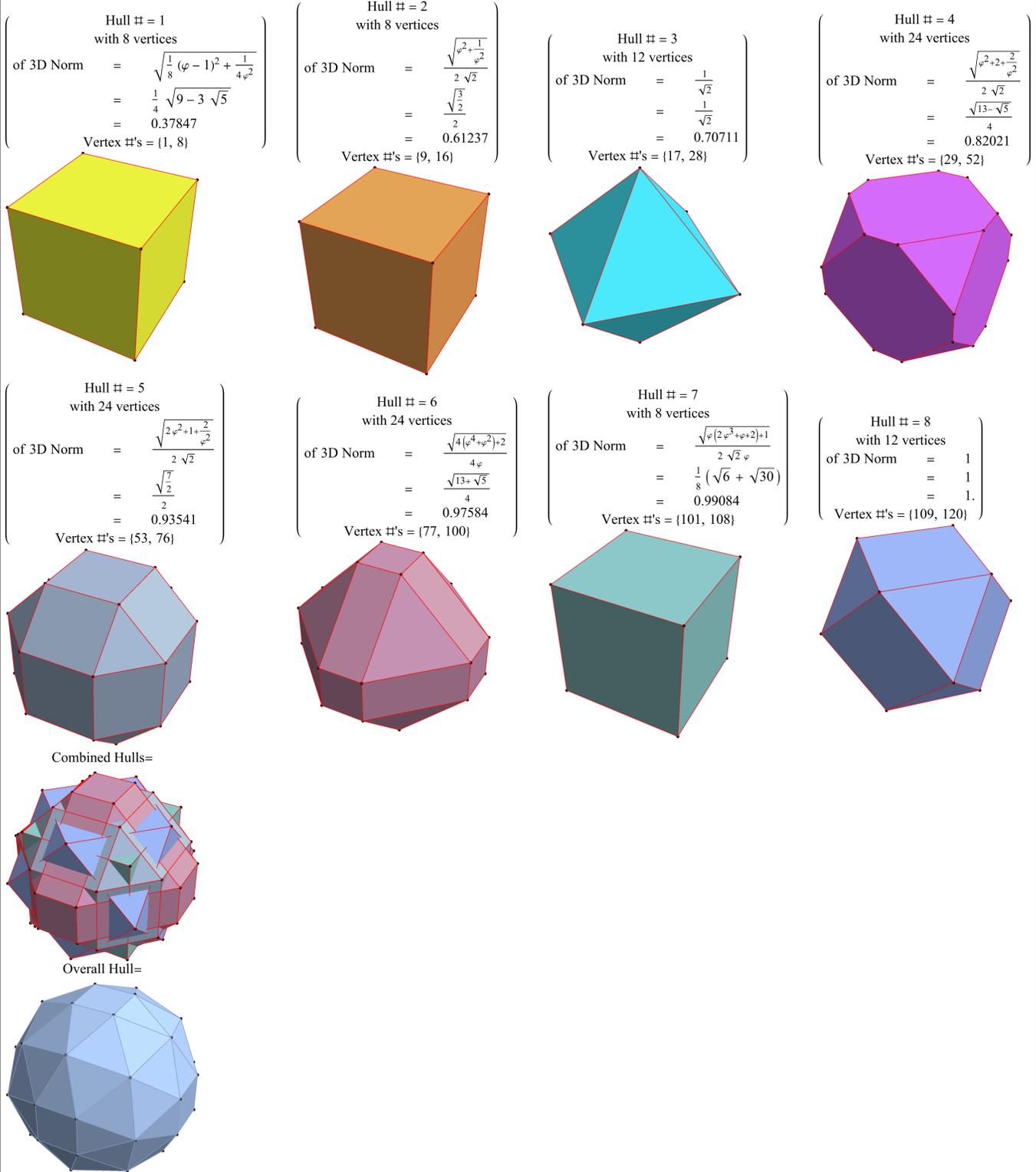


FIG. 15. Concentric hulls of I' as the parent H_4 600-cell of order 120 in Platonic 3D projection with numeric and symbolic norm distances. This is generated by $I' = \text{prq}[\alpha^{0-4}, 1, T']$.

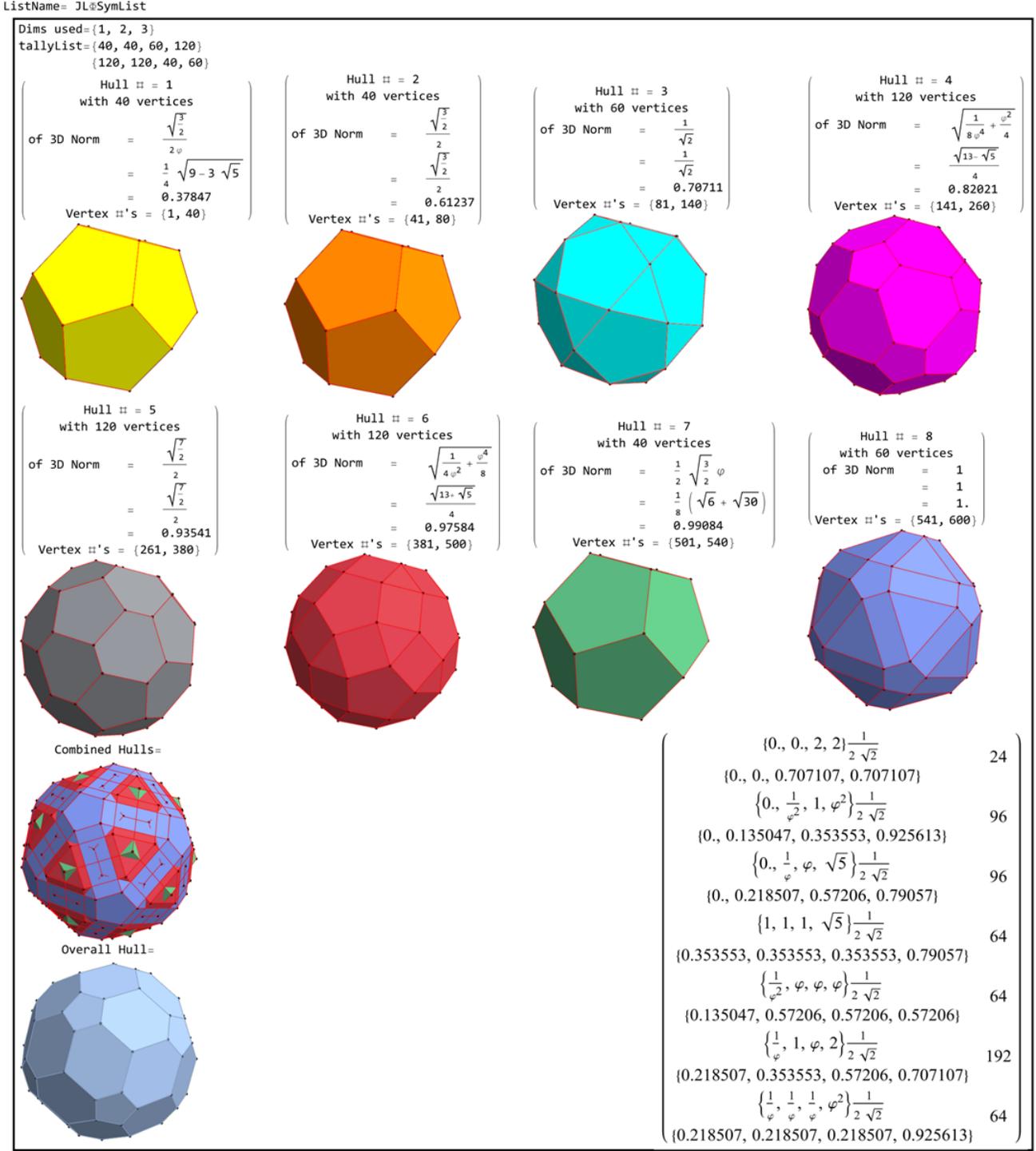


FIG. 16. Concentric hulls of J as the tri-rectified H_4 120-cell of order 600 in Platonic 3D projection with numeric and symbolic norm distances. This is generated by $J = \text{prq}[A', 1, I] = \text{prq}[A', \alpha^{0-4}, T]$. Note: The numeric and symbolic tally list of unpermuted vertex values in the lower-right corner

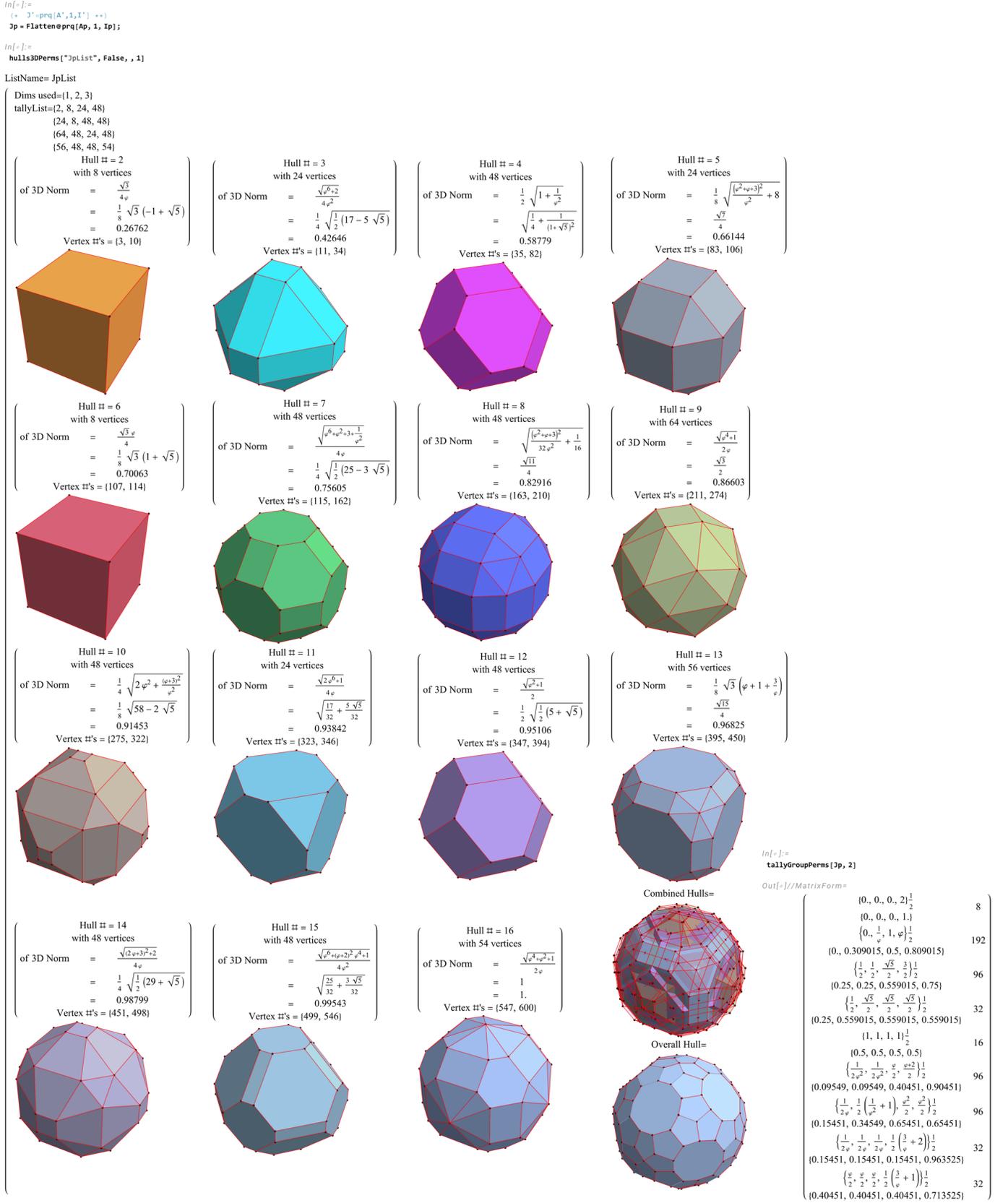


FIG. 17. Concentric hulls of J' as the tri-rectified H_4 120-cell of order 600 in Platonic 3D projection with numeric and symbolic norm distances. This is generated by $J' = \text{prq}[A', 1, I'] = \text{prq}[A', \beta^{0-4}, T']$. Note: The numeric and symbolic tally list of unpermuted vertex values in the lower-right corner

Group	Archimedean Solid	Dual Catalan Solid	Irregular Solid	Chiral Solid
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				
H_3				
A_3				
B_3				

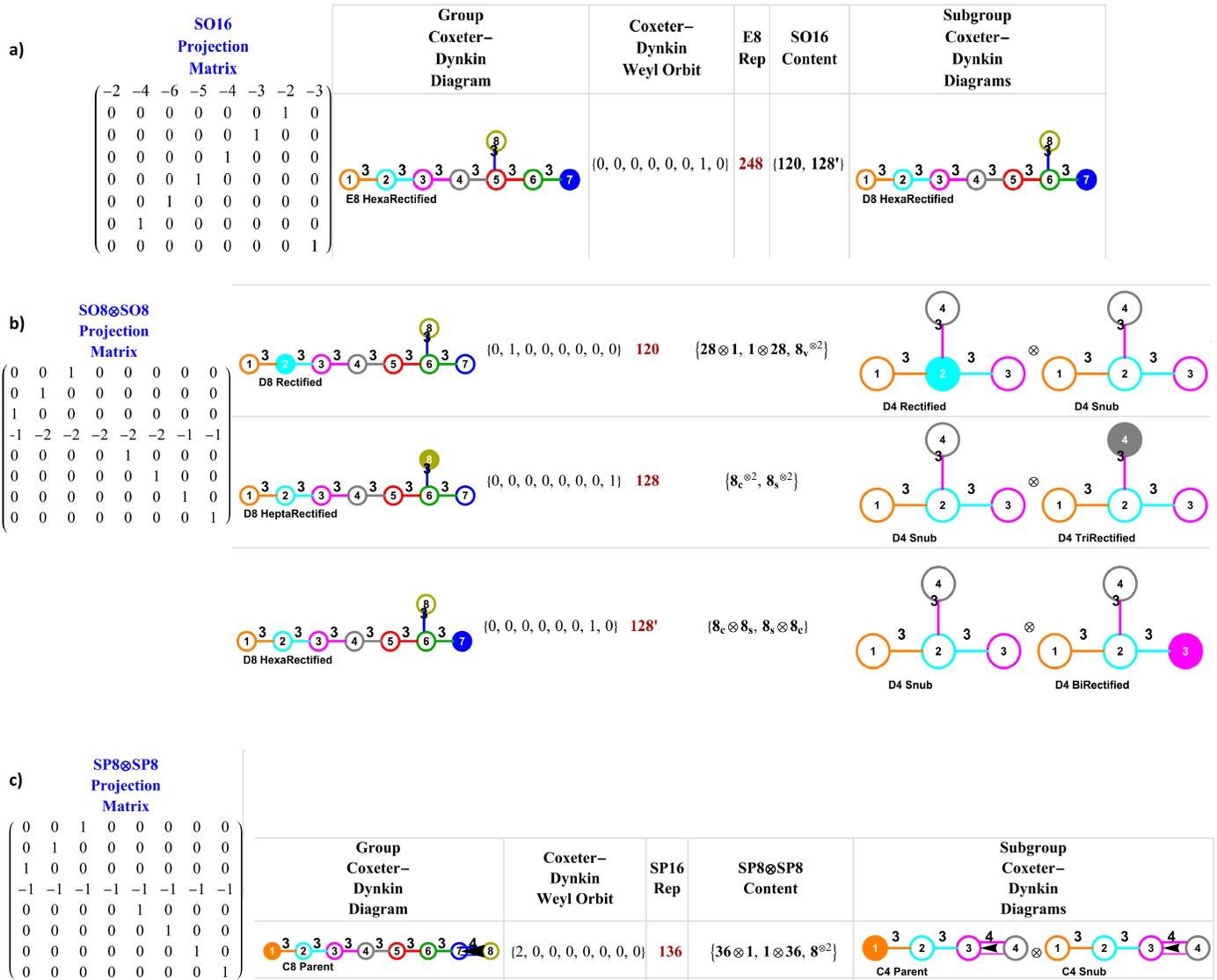


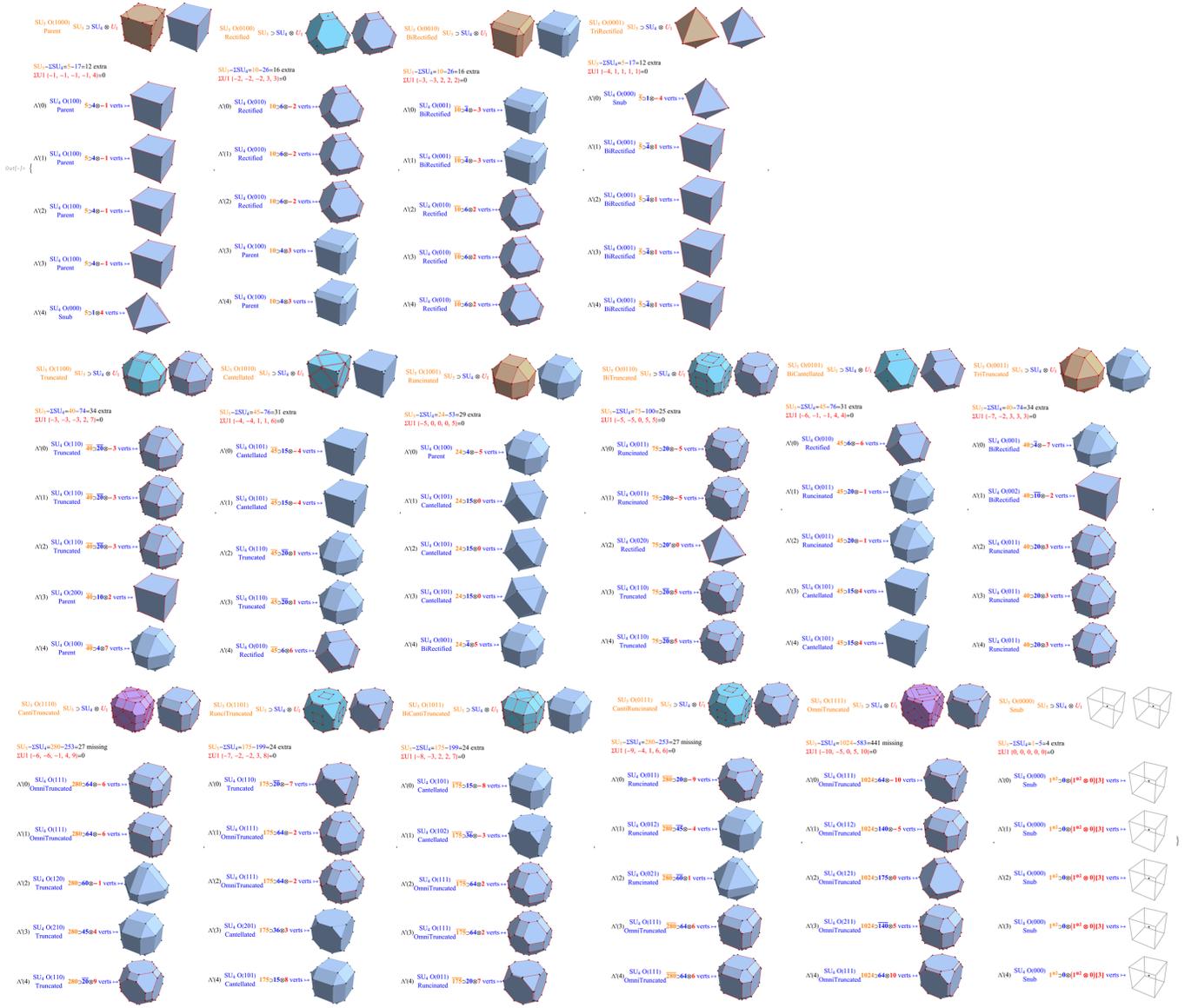
FIG. 19. Breakdown of E_8 maximal embeddings at height 248 of content $SO(16)=D_8$ (120,128')

a) Height 248 $SO(16)$ content $120=(112+4+4)+128'$

b) Height 120 and 128' $SO(8) \otimes SO(8)$ content w/ $8_{v,c,s}^{\otimes 2}$ triality

c) Height 136 $Sp(8) \otimes Sp(8)$ content $(32+4) \otimes 1, 1 \otimes (32+4), 8^{\otimes 2}$

Note: This output was created in *Mathematica*TM with support from the GroupMath[21] and SuperLie[22] packages.



```

In[*]:= (* This switches the H4 (L)eft side scale to the (R)ight side scale (and vice-versa).
We don't use scaleBy if it is a snub 24-cell vertices. *)
switchScale[in_, scaleBy_ : 1] := (* We don't use scaleBy if it is a snub 24-cell vertices. *)
If[Length@Union@Flatten@Abs@oct2List@N[in /. φRep] == 2,
in, scaleBy in /. slRep];

In[*]:= (* Replacement order is critical *)
mapLRrep = # /. slRep & /@ {
(* φ±3 Scale changing: Exchange the ±φ2↔±1/φ and ±φ±3/2↔±φ∓3/2 *)

$$\frac{1}{\phi SW^2} \rightarrow \varphi, \phi SW^2 \rightarrow \frac{1}{\varphi}, \phi SW^{-3/2} \rightarrow \varphi^{3/2}, \phi SW^{3/2} \rightarrow \varphi^{-3/2},$$

(* Sign changing: Exchange the ±√φ↔∓√φ & ±1/√φ↔∓1/√φ, and ±1/φ↔∓1/φ *)

$$\sqrt{\phi SW} \rightarrow -\sqrt{\varphi}, \sqrt{\frac{1}{\phi SW}} \rightarrow -\sqrt{\frac{1}{\varphi}}, \frac{1}{\phi SW} \rightarrow -\frac{1}{\varphi},$$

(* Final φ±3 Scale changing: ±φ↔±1/φ2 *)

$$\phi SW \rightarrow \frac{1}{\varphi^2} (**);$$

};

In[*]:= (* This processes only individual vertices with a symbolic list input. *)
mapLR[in_, scaleBy_ : 1, uDetIfCorrection_ : True] := Module[{(*input,output*)},
(* Correct for use of √φ in U which produces i values (which may be desired?) *)
input = If[currU == 11 || ! uDetIfCorrection, in, FullSimplify[in uDetIf /. slRep, Assumptions → φAssumptions]];
output = FullSimplify[switchScale[octSym@input /. φ → φSW /. mapLRrep /. slRep, scaleBy] ×
(* Correct back *)
If[currU == 11 || ! uDetIfCorrection, 1, 1 / uDetIf] /. slRep, Assumptions → φAssumptions] /. slRep;
(* currU < 9 don't reverse the L↔R ordering *)
If[currU < 9, output, Join[Reverse[output[ ; 4]], Array[0 &, Length[output] - 4]]];

(* List and verify the operation of mapLR - one for h4⊕ and one for h4 *)
genE8fromH4@in_String := Module[{indx, inH4 = If[in == "H4⊕", h4⊕, h4], i, j, left, right, h4LR},
(* Style the Heading in Bold, 24-cell rows in Red, and p48 constraint members marked with an * *)
Style[#,
{If[MemberQ[If[in == "H4⊕", h4⊕cell124, h4cell124], indx], Red, Black],
If[Head@indx === String, Bold, Plain]}] & /@ (indx = #[[2]; #) & /@
Join[
(* The Heading row *)
{{"#", in <> " #", If[labels, "pLb1", Nothing],
Column[{"E8 vertex", "E8.U=" <> in <> " L" <> "⊕" <> in <> " R"}, Center],
Column[If[currU == 11, "", "2 "] <> in <> " L", "mapLR(" <> in <> " L)=" <> in <> " R"}, Center],
Column[If[currU == 11, "", "2 "] <> in <> " R", "mapLR(" <> in <> " R)=" <> in <> " L"}, Center],
Column[{"", "(" <> in <> " L" <> "⊕" <> in <> " R" <> ") .U-1=E8 vertex"}, Center],
Column[{"E8→" <> in <> " L" <> "⊕" <> in <> " R" <> "≡", in <> " L" <> "⊕" <> in <> " R" <> "→E8"}, Center]}}],
(* Generate data row content *)
{ToString@# <> If[MemberQ[If[in == "H4⊕", p48L⊕, p48L], #], "*", " "],
(* h4⊕[#] is an E8 index number to an E8 element in h4⊕ *)
inH4[#], If[labels, pLb1@inH4[#], Nothing],
(* Show the E8 vertex *)
i = pE8@inH4[#],
(* pC600 is converts from E8→H4 using U, here we take the H4 4D left side *)
If[currU == 11, 1, 2] (left = octSym[pC600[inH4[#]]][ ; 4]) /. φRuleList),
(* mapLR converts the H4 4D left side vertex to its corresponding H4 4D right-side vertex,
which when Joined gives the 8D H4 that can be converted back to E8 by using UInv *)
If[currU == 11, 1, 2] (right = mapLR@left /. φRuleList),
(* Conditionally print some cross-checks *)
print["#=", #, " h4[#]=", inH4[#], " E8.U=", octSym[pC600[inH4[#]]] /. φRuleList, " left=",
left, " right=", Reverse@right];
print[" E8.U==Join[left,mapLR@left]=", N@Join[left, right] == N@octSym[pC600[inH4[#]]] /. φRuleList];
print[" mapLR@right", If[currU == 11, 1, 2] (mapLR@right /. φRuleList)];
print[" left==mapLR@right=", N@left == N@mapLR@right /. φRuleList];
(* Show the H4L@H4R.UInv vertex *)
h4LR = Join[left, right];
j = Rationalize@FullSimplify[Chop[h4LR.UInv /. φRep, chop], Assumptions → φAssumptions],
(* Check that E8→H4→H4L@H4R→E8 *)
j == N@i} /. slRep & /@ Range@120] // MatrixForm];

```

FIG. 21. *Mathematica*TM code to generate the output showing $E_8 \leftrightarrow H_4$ isomorphism

in[]:= curru = 9; setf; (octSym# => mapLR# & . slRep) & /o { { [1, tau, 0, -1, 1/tau^2], [-1/tau, 0, 1, -1/tau^2], {tau, 0, -1, 1/tau}, [-tau, 0, 1, -1/tau]} / If[curru = 11, 1, uDetif] // slRep // MatrixForm (+ 24-cell rows in Red and p48 constraint members marked with an *) genE8fromH4@H4"		out[]:=MatrixForm= { {1/sqrt(2), 0, -sqrt(2)/2, 1/(2*sqrt(2))} - {sqrt(2)/2, 0, -1/(2*sqrt(2))}, {-1/(2*sqrt(2)), 0, sqrt(2)/2, -1/(2*sqrt(2))} - {sqrt(2)/2, 0, 1/(2*sqrt(2))}, {sqrt(2)/2, 0, -sqrt(2)/2, 1/(2*sqrt(2))} - {sqrt(2)/2, 0, 1/(2*sqrt(2))}, {sqrt(2)/2, 0, -sqrt(2)/2, -1/(2*sqrt(2))} - {sqrt(2)/2, 0, -1/(2*sqrt(2))}}											
#	H4 #	E8 vertex E8L@H4_g@H4_g	2 H4_L mapLR(H4_L)=H4_g	2 H4_g mapLR(H4_g)=H4_L	(H4_L@H4_g)U^1=E8 vertex	E8-H4_L@H4_g= H4_L@H4_g->E8	#	H4 #	E8 vertex E8L@H4_g@H4_g	2 H4_L mapLR(H4_L)=H4_g	2 H4_g mapLR(H4_g)=H4_L	(H4_L@H4_g)U^1=E8 vertex	E8-H4_L@H4_g= H4_L@H4_g->E8
1	13	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), -sqrt(2), 0}	{0, -sqrt(2), -1/sqrt(2)}	{1/2, -1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True	61	129	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{sqrt(2), -1/sqrt(2), 0}	{sqrt(2), 0, 1/sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True
2	14	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), 0, -sqrt(2)}	{-sqrt(2), -sqrt(2), 0}	{1/2, -1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True	62	130	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{sqrt(2), sqrt(2), -sqrt(2)}	{0, sqrt(2), sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True
3	15	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), -sqrt(2), 0, -sqrt(2)}	{-sqrt(2), 0, -sqrt(2), -1/sqrt(2)}	{1/2, -1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True	63	131	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{sqrt(2), 0, 1/sqrt(2), -1/sqrt(2)}	{1/sqrt(2), sqrt(2), 0, sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True
4	20	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), -1/sqrt(2), 0, -sqrt(2)}	{-sqrt(2), 0, 1/sqrt(2), sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True	64	132	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), -1/sqrt(2), -1/sqrt(2), -1/sqrt(2)}	{1/sqrt(2), 1/sqrt(2), 1/sqrt(2), 1/sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True
5	24	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), 0, -sqrt(2), -1/sqrt(2)}	{sqrt(2), 0, -sqrt(2), 0, sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True	65	133	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), sqrt(2), 1/sqrt(2), 0}	{0, -sqrt(2), sqrt(2), sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True
6	30	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), -sqrt(2), 0}	{0, -sqrt(2), sqrt(2), sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True	66	135	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), -1/sqrt(2), -1/sqrt(2), -1/sqrt(2)}	{1/sqrt(2), -1/sqrt(2), 1/sqrt(2), 1/sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True
7	32	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), 0, -1/sqrt(2), -1/sqrt(2)}	{-1/sqrt(2), -sqrt(2), 0, sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True	67	137	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{-1/sqrt(2), sqrt(2), 0, -sqrt(2)}	{-sqrt(2), 0, sqrt(2), sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True
8	33	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), -sqrt(2), 1/sqrt(2), 0}	{0, -1/sqrt(2), sqrt(2), sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True	68	142	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), 0, 1/sqrt(2), -1/sqrt(2)}	{-1/sqrt(2), sqrt(2), 0, sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True
9	34	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), -sqrt(2), -1/sqrt(2), 0}	{0, -sqrt(2), -sqrt(2), 0}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True	69	143	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{-1/sqrt(2), -1/sqrt(2), -1/sqrt(2), -1/sqrt(2)}	{-1/sqrt(2), -1/sqrt(2), 1/sqrt(2), 1/sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True
10	35	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), 1/sqrt(2), 0, -1/sqrt(2)}	{-sqrt(2), 0, -1/sqrt(2), -1/sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True	70	144	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), -sqrt(2), 1/sqrt(2), 0}	{0, sqrt(2), -sqrt(2), sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True
11	36	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), 0, -sqrt(2), -1/sqrt(2)}	{-sqrt(2), -sqrt(2), 0, 1/sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True	71	146	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{-1/sqrt(2), -1/sqrt(2), 1/sqrt(2), 1/sqrt(2)}	{-1/sqrt(2), -1/sqrt(2), -1/sqrt(2), -1/sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True
12	37	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), -1/sqrt(2), 0, -sqrt(2)}	{-sqrt(2), 0, -sqrt(2), 1/sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True	72	148	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{-1/sqrt(2), 0, sqrt(2), -1/sqrt(2)}	{-sqrt(2), sqrt(2), 0, 1/sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True
13	38	{-1, -1, 0, 0, 0, 0, 0, 0}	{1/sqrt(2), -sqrt(2), 0}	{0, -sqrt(2), -sqrt(2), sqrt(2)}	{-1, -1, 0, 0, 0, 0, 0, 0}	True	73	150	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), 0, 1/sqrt(2), -1/sqrt(2)}	{sqrt(2), 0, -1/sqrt(2), sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True
14	39	{-1, 0, 0, -1, 0, 0, 0, 0}	{1/sqrt(2), 0, -sqrt(2), 1/sqrt(2)}	{-1/sqrt(2), 0, -sqrt(2), sqrt(2)}	{-1, 0, 0, -1, 0, 0, 0, 0}	True	74	151	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{-1/sqrt(2), 0, -sqrt(2), 1/sqrt(2)}	{sqrt(2), -sqrt(2), 0, 1/sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True
15	40	{-1, 0, 0, 0, -1, 0, 0, 0}	{1/sqrt(2), 0, 1/sqrt(2), -sqrt(2)}	{-sqrt(2), -1/sqrt(2), 0, sqrt(2)}	{-1, 0, 0, 0, -1, 0, 0, 0}	True	75	154	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{-1/sqrt(2), 1/sqrt(2), -1/sqrt(2), -1/sqrt(2)}	{1/sqrt(2), 1/sqrt(2), 1/sqrt(2), 1/sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True
16	41	{-1, 0, 0, 0, 0, -1, 0, 0}	{1/sqrt(2), 0, -1/sqrt(2), -sqrt(2)}	{-sqrt(2), 1/sqrt(2), 0, sqrt(2)}	{-1, 0, 0, 0, 0, -1, 0, 0}	True	76	156	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{-1/sqrt(2), sqrt(2), 0, -sqrt(2)}	{0, -sqrt(2), sqrt(2), -sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True
17	42	{-1, 0, 0, 0, 0, 0, -1, 0, 0}	{1/sqrt(2), -sqrt(2), 0, 1/sqrt(2)}	{0, -sqrt(2), -sqrt(2), sqrt(2)}	{-1, 0, 0, 0, 0, -1, 0, 0}	True	77	157	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{-1/sqrt(2), 1/sqrt(2), -1/sqrt(2), -1/sqrt(2)}	{1/sqrt(2), 1/sqrt(2), 1/sqrt(2), 1/sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True
18	43	{-1, 0, 0, 0, 0, 0, 0, -1, 0}	{1/sqrt(2), -1/sqrt(2), -sqrt(2), 0}	{0, -sqrt(2), -sqrt(2), sqrt(2)}	{-1, 0, 0, 0, 0, 0, -1, 0}	True	78	161	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{-1/sqrt(2), -sqrt(2), 0, sqrt(2)}	{sqrt(2), 0, -sqrt(2), sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True
19	45	{-1, 0, 0, 0, 0, 0, 0, 0, 1}	{-2*sqrt(2), 0, 0, 0}	{0, 0, 0, 0, 1/sqrt(2)}	{-1, 0, 0, 0, 0, 0, 0, 1}	True	79	162	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{-1/sqrt(2), 1/sqrt(2), -1/sqrt(2), -1/sqrt(2)}	{-1/sqrt(2), 1/sqrt(2), 1/sqrt(2), 1/sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True
20	46	{-1, 0, 0, 0, 0, 0, 0, 0, 1}	{1/sqrt(2), -sqrt(2), sqrt(2), 0}	{0, sqrt(2), -sqrt(2), sqrt(2)}	{-1, 0, 0, 0, 0, 0, 0, 1}	True	80	163	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{-1/sqrt(2), -1/sqrt(2), 1/sqrt(2), 1/sqrt(2)}	{-1/sqrt(2), -1/sqrt(2), -1/sqrt(2), -1/sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True
21	47	{-1, 0, 0, 0, 0, 0, 0, 0, 1}	{1/sqrt(2), sqrt(2), 0, 1/sqrt(2)}	{1/sqrt(2), 0, sqrt(2), sqrt(2)}	{-1, 0, 0, 0, 0, 0, 0, 1}	True	81	166	{0, 0, 0, 0, 0, 1, -1, 0}	{0, sqrt(2), -sqrt(2), 0}	{-sqrt(2), -sqrt(2), 0, 0}	{0, 0, 0, 0, 0, 1, -1, 0}	True
22	48	{-1, 0, 0, 0, 0, 0, 0, 0, 1}	{1/sqrt(2), 0, 1/sqrt(2), sqrt(2)}	{sqrt(2), -1/sqrt(2), 0, sqrt(2)}	{-1, 0, 0, 0, 0, 0, 0, 1}	True	82	170	{0, 0, 0, 0, 1, -1, 0, 0}	{0, -sqrt(2), 1/sqrt(2), 1/sqrt(2)}	{sqrt(2), -1/sqrt(2), -sqrt(2), 0}	{0, 0, 0, 0, 1, -1, 0, 0}	True
23	49	{-1, 0, 0, 0, 0, 0, 0, 0, 1}	{1/sqrt(2), 0, -1/sqrt(2), sqrt(2)}	{sqrt(2), 1/sqrt(2), 0, sqrt(2)}	{-1, 0, 0, 0, 0, 0, 0, 1}	True	83	171	{0, 0, 0, 0, 1, 0, -1, 0}	{0, -sqrt(2), -sqrt(2), sqrt(2)}	{sqrt(2), -sqrt(2), 1/sqrt(2), 0}	{0, 0, 0, 0, 1, 0, -1, 0}	True
24	50	{-1, 0, 0, 0, 0, 0, 0, 0, 1}	{1/sqrt(2), sqrt(2), 0, -1/sqrt(2)}	{0, sqrt(2), sqrt(2), sqrt(2)}	{-1, 0, 0, 0, 0, 0, 0, 1}	True	84	176	{0, 0, 0, 1, -1, 0, 0, 0}	{0, 0, -2*sqrt(2), 0}	{0, sqrt(2), 0, 0}	{0, 0, 0, 1, -1, 0, 0, 0}	True
25	51	{-1, 0, 0, 0, 0, 0, 0, 0, 1}	{1/sqrt(2), -sqrt(2), 0, 1/sqrt(2)}	{0, sqrt(2), sqrt(2), sqrt(2)}	{-1, 0, 0, 0, 0, 0, 0, 1}	True	85	177	{0, 0, 0, 1, 0, 0, -1, 0}	{0, -sqrt(2), -1/sqrt(2), sqrt(2)}	{sqrt(2), 1/sqrt(2), -sqrt(2), 0}	{0, 0, 0, 1, 0, 0, -1, 0}	True
26	52	{0, -1, -1, 0, 0, 0, 0, 0, 0}	{0, -1/sqrt(2), -sqrt(2), 1/sqrt(2)}	{-1/sqrt(2), -sqrt(2), 0}	{0, -1, -1, 0, 0, 0, 0, 0}	True	86	181	{0, 0, 0, 1, 0, 0, 0, 0}	{0, 1/sqrt(2), sqrt(2), sqrt(2)}	{sqrt(2), sqrt(2), -1/sqrt(2), 0}	{0, 0, 0, 1, 0, 0, 0, 0}	True
27	53	{0, -1, 0, -1, 0, 0, 0, 0, 0}	{0, 1/sqrt(2), -sqrt(2), -sqrt(2)}	{-sqrt(2), -sqrt(2), -1/sqrt(2), 0}	{0, -1, 0, -1, 0, 0, 0, 0}	True	87	186	{0, 0, 0, 1, 0, 0, -1, 0}	{0, 0, 0, -2*sqrt(2), 0}	{1/sqrt(2), 0, 0, 0}	{0, 0, 0, 1, 0, 0, -1, 0}	True
28	55	{0, -1, 0, 0, 0, -1, 0, 0, 0}	{0, -1/sqrt(2), -sqrt(2), -1/sqrt(2)}	{-1/sqrt(2), -sqrt(2), 0}	{0, -1, 0, 0, 0, -1, 0, 0}	True	88	187	{0, 0, 1, 0, 0, 0, -1, 0}	{0, sqrt(2), -sqrt(2), -1/sqrt(2)}	{-1/sqrt(2), -sqrt(2), sqrt(2), 0}	{0, 0, 1, 0, 0, 0, -1, 0}	True
29	59	{0, -1, 0, 0, 0, 0, 0, 0, 1}	{0, 2*sqrt(2), 0, 0}	{0, 0, 0, 0, 1/sqrt(2)}	{0, -1, 0, 0, 0, 0, 0, 1}	True	89	192	{0, 0, 1, 0, 1, 0, 0, 0}	{0, sqrt(2), 1/sqrt(2), sqrt(2)}	{sqrt(2), 1/sqrt(2), -sqrt(2), 0}	{0, 0, 1, 0, 1, 0, 0, 0}	True
30	61	{0, -1, 0, 0, 0, 0, 0, 0, 1}	{0, 1/sqrt(2), -sqrt(2), sqrt(2)}	{sqrt(2), -sqrt(2), -1/sqrt(2), 0}	{0, -1, 0, 0, 0, 0, 0, 1}	True	90	193	{0, 0, 0, 1, 0, 0, 0, 0}	{0, -sqrt(2), -1/sqrt(2), sqrt(2)}	{sqrt(2), 1/sqrt(2), sqrt(2), 0}	{0, 0, 0, 1, 0, 0, 0, 0}	True
31	64	{0, 0, -1, -1, 0, 0, 0, 0, 0}	{0, -sqrt(2), 1/sqrt(2), -sqrt(2)}	{-sqrt(2), -1/sqrt(2), -sqrt(2), 0}	{0, 0, -1, -1, 0, 0, 0, 0}	True	91	196	{0, 1, 0, 0, -1, 0, 0, 0}	{0, -1/sqrt(2), 1/sqrt(2), -sqrt(2)}	{-sqrt(2), sqrt(2), 1/sqrt(2), 0}	{0, 1, 0, 0, -1, 0, 0, 0}	True
32	65	{0, 0, -1, 0, -1, 0, 0, 0, 0}	{0, -sqrt(2), -1/sqrt(2), -sqrt(2)}	{-sqrt(2), 1/sqrt(2), -sqrt(2), 0}	{0, 0, -1, 0, -1, 0, 0, 0}	True	92	198	{0, 1, 0, 0, 0, 0, -1, 0}	{0, -2*sqrt(2), 0, 0}	{0, 0, 1/sqrt(2), 0}	{0, 1, 0, 0, 0, 0, -1, 0}	True
33	70	{0, 0, -1, 0, 0, 0, 0, 0, 1}	{0, 1/sqrt(2), sqrt(2), -1/sqrt(2)}	{-1/sqrt(2), sqrt(2), -sqrt(2), 0}	{0, 0, -1, 0, 0, 0, 0, 1}	True	93	202	{0, 1, 0, 0, 0, 0, 0, 0}	{0, 1/sqrt(2), sqrt(2), -1/sqrt(2)}	{-1/sqrt(2), sqrt(2), sqrt(2), 0}	{0, 1, 0, 0, 0, 0, 0, 0}	True
34	71	{0, 0, -1, 0, 0, 0, 0, 0, 1}	{0, 0, 0, 2*sqrt(2), 0}	{-1/sqrt(2), 0, 0, 0}	{0, 0, -1, 0, 0, 0, 0, 1}	True	94	204	{0, 1, 0, 1, 0, 0, 0, 0}	{0, -1/sqrt(2), sqrt(2), sqrt(2)}	{sqrt(2), sqrt(2), 1/sqrt(2), 0}	{0, 1, 0, 1, 0, 0, 0, 0}	True
35	76	{0, 0, 0, -1, 0, 0, -1, 0, 0}	{0, -1/sqrt(2), -sqrt(2), -sqrt(2)}	{-sqrt(2), -sqrt(2), 1/sqrt(2), 0}	{0, 0, 0, -1, 0, 0, -1, 0}	True	95	205	{0, 1, 1, 0, 0, 0, 0, 0}	{0, 1/sqrt(2), sqrt(2), -1/sqrt(2)}	{1/sqrt(2), sqrt(2), sqrt(2), 0}	{0, 1, 1, 0, 0, 0, 0, 0}	True
36	80	{0, 0, 0, -1, 0, 0, 0, 0, 0}	{0, 1/sqrt(2), -sqrt(2), -sqrt(2)}	{-sqrt(2), -sqrt(2), sqrt(2), 0}	{0, 0, 0, -1, 0, 0, 0, 0}	True	96	206	{1, -1, 0, 0, 0, 0, 0, 0}	{-1/sqrt(2), 1/sqrt(2), -sqrt(2), 0}	{0, -sqrt(2), -1/sqrt(2), -sqrt(2)}	{1, -1, 0, 0, 0, 0, 0, 0}	True
37	81	{0, 0, 0, -1, 0, 0, 0, 0, 0}	{0, 0, 2*sqrt(2), 0}	{0, -sqrt(2), 0, 0}	{0, 0, 0, -1, 0, 0, 0, 0}	True	97	207	{1, 0, -1, 0, 0, 0, 0, 0}	{-1/sqrt(2), -sqrt(2), 0, 1/sqrt(2)}	{-1/sqrt(2), 0, -sqrt(2), -sqrt(2)}	{1, 0, -1, 0, 0, 0, 0, 0}	True
38	86	{0, 0, 0, 0, -1, 0, 0, 0, 0}	{0, 1/sqrt(2), sqrt(2), -sqrt(2)}	{-sqrt(2), sqrt(2), -1/sqrt(2), 0}	{0, 0, 0, 0, -1, 0, 0, 0}	True	98	208	{1, 0, 0, -1, 0, 0, 0, 0}	{-1/sqrt(2), 0, 1/sqrt(2), -sqrt(2)}	{-sqrt(2), -1/sqrt(2), 0, -sqrt(2)}	{1, 0, 0, -1, 0, 0, 0, 0}	True
39	87	{0, 0, 0, 0, 0, -1, 0, 0, 0}	{0, sqrt(2), -1/sqrt(2), -sqrt(2)}	{sqrt(2), 1/sqrt(2), -sqrt(2), 0}	{0, 0, 0, 0, -1, 0, 0, 0}	True	99	209	{1, 0, 0, 0, -1, 0, 0, 0}	{-1/sqrt(2), 0, -sqrt(2), -sqrt(2)}	{-sqrt(2), -sqrt(2), 0, -sqrt(2)}	{1, 0, 0, 0, -1, 0, 0, 0}	True
40	91	{0, 0, 0, 0, 0, -1, 0, 0, 0}	{0, -sqrt(2), sqrt(2), -1/sqrt(2)}	{1/sqrt(2), sqrt(2), -sqrt(2), 0}	{0, 0, 0, 0, -1, 0, 0, 0}	True	100	210	{1, 0, 0, 0, 0, -1, 0, 0}	{-1/sqrt(2), -sqrt(2), 0, -1/sqrt(2)}	{1/sqrt(2), 0, -sqrt(2), -sqrt(2)}	{1, 0, 0, 0, 0, -1, 0, 0}	True
41	94	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), -1/sqrt(2), -1/sqrt(2), -1/sqrt(2)}	{-1/sqrt(2), 1/sqrt(2), -1/sqrt(2), -1/sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True	101	211	{1, 0, 0, 0, 0, 0, -1, 0}	{-1/sqrt(2), -1/sqrt(2), -sqrt(2), 0}	{0, -sqrt(2), 1/sqrt(2), -sqrt(2)}	{1, 0, 0, 0, 0, 0, -1, 0}	True
42	95	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), 1/sqrt(2), -1/sqrt(2), -1/sqrt(2)}	{-sqrt(2), -sqrt(2), 1/sqrt(2), -1/sqrt(2)}	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	True	102	212	{1, 0, 0, 0, 0, 0, -1, 0}	{2*sqrt(2), 0, 0, 0}	{0, 0, 0, 0, -1/sqrt(2)}	{1, 0, 0, 0, 0, 0, -1, 0}	True
43	96	{1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2}	{1/sqrt(2), -sqrt(2),										

