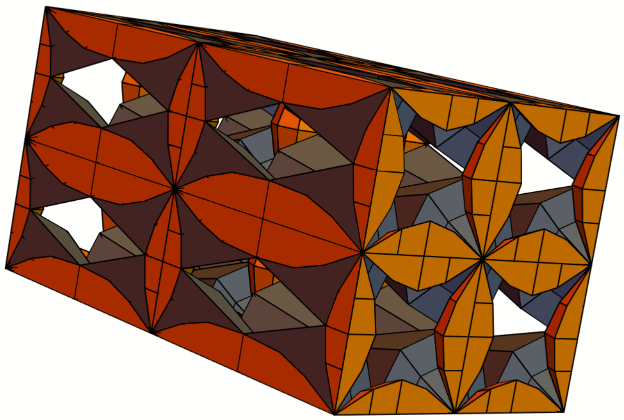
Construction Of 3D Logo



We start with a polygon giving my 2D logo

```
In[2]:= corb2D := {{0.5`, 0.22`}, {0.28`, 0.16`}, {0, 0}, {0.15`, 0.33`}, {0.23`, 0.61`}, {0.25`, 1}}
```

This will be the basis for the entire construction. We make it dynamic by

```
In[3]:= Dcorb2D[α_] := Polygon[α@σ2D /@ corb2D]
```

In two dimensions we need to use combinations of the following transformations, a rotation, a translation and a reflection. We will see that in 3D we will not need the reflection.

We have the graphic

```
In[4]:= σ2D = RotationTransform[-Pi / 2, {0, 0}]
```

Out[4]= TransformationFunction$\left[\left(\begin{array}{cc|c} 0 & 1 & 0 \\ -1 & 0 & 0 \\ \hline 0 & 0 & 1 \end{array}\right)\right]$

```
In[5]:= τ2Dx = TranslationTransform[{2, 0}]
```

Out[5]= TransformationFunction$\left[\left(\begin{array}{cc|c} 1 & 0 & 2 \\ 0 & 1 & 0 \\ \hline 0 & 0 & 1 \end{array}\right)\right]$

reflectionTransform

```
In[6]:= ρ2D = TransformationFunction[{{-1, 0, 0}, {0, 1, 0}, {0, 0, 1}}]
```

Out[6]= TransformationFunction$\left[\left(\begin{array}{cc|c} -1 & 0 & 0 \\ 0 & 1 & 0 \\ \hline 0 & 0 & 1 \end{array}\right)\right]$

```
In[7]:= τ2Dy = TranslationTransform[{0, 1}]
```

Out[7]= TransformationFunction$\left[\left(\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 1 & 1 \\ \hline 0 & 0 & 1 \end{array}\right)\right]$

Also we always need the identity, do nothing transform

In[8]:=
```
ι2 = TranslationTransform[{0, 0}]
```

Out[8]=
```
TransformationFunction[ ( 1 0 | 0
                          0 1 | 0
                          -------
                          0 0 | 1 ) ]
```

A half turn is useful

In[9]:=
```
η2Da := RotationTransform[Pi, {1, -.5}]
```

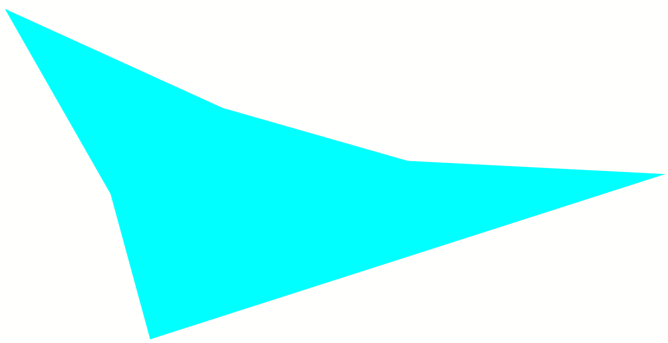
Now we can define our first dynamic object. Notice we rotate to get a better view point.

In[10]:=
```
Dcorb2D[α_] := Polygon[α@*σ2D /@ corb2D]
```

The graphic is

In[11]:=
```
Graphics[{Cyan, Dcorb2D[ι2]}]
```

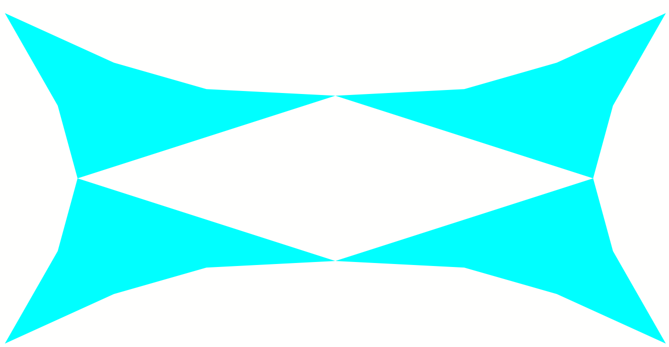
Out[11]=



We construct the main part of my 2D logo

In[12]:=
```
Graphics[
 {{Cyan, Dcorb2D[ι2], Dcorb2D[η2Da], Dcorb2D[τ2Dx@*ρ2D], Dcorb2D[η2Da@*τ2Dx@*ρ2D]}}]
```

Out[12]=



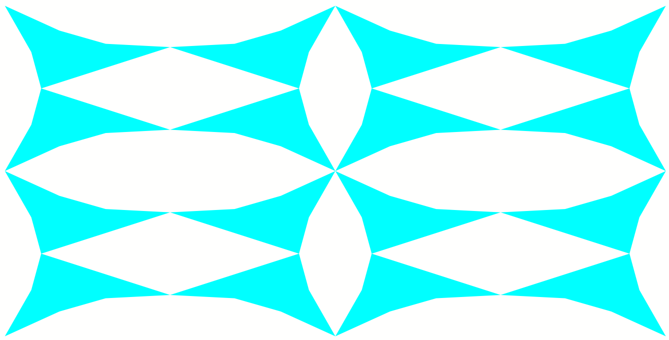
We want to repeat it so we make this dynamic

In[13]:=
```
LogoCell[β_] :=
 {Dcorb2D[β], Dcorb2D[β@*η2Da], Dcorb2D[β@*τ2Dx@*ρ2D], Dcorb2D[β@*η2Da@*τ2Dx@*ρ2D]}
```

This is easier to repeat, we can do

In[14]:=
```
Graphics[{Cyan, LogoCell[ι2], LogoCell[τ2Dx], LogoCell[τ2Dy], LogoCell[τ2Dy@⋆τ2Dx]}]
```

Out[14]=



which is a start on my 2D logo which comprises the white spaces.

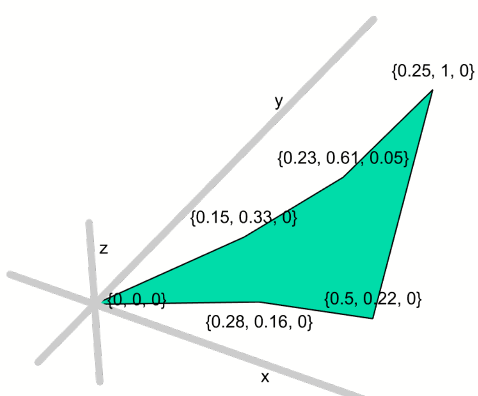
Now we go to 3D. We first embed our 2 dimensional basic cell in 3D adding a zero to each point.

In[15]:=
```
corb := {{.5, .22, 0}, {.28, .16, 0}, {0, 0, 0}, {.15, .33, 0}, {.23, .61, 0}, {.25, 1, 0}}
```

The graphic is

In[16]:=
```
Graphics3D[
 {{Cyan, Polygon[corb]}, {GrayLevel[.8], Thickness[.01], Line[{{0, 0, -.2}, {0, 0, .2}}],
   Line[{{0, -.2, 0}, {0, 1.2, 0}}], Line[{{-.2, 0, 0}, {.6, 0, 0}}]},
  {Black, Inset["x", {.4, -.03, .0}], Inset[{0, 0, 0}, {.1, 0, 0.05}],
   Inset["y", {-.03, .8, 0}], Inset["z", {0.03, 0, .15}],
   Inset[{.15, .33, 0}, {.15, .33, 0.05}], Inset[{.23, .61, 0.05}, {.23, .61, .05}],
   Inset[{.25, 1, 0}, {.25, 1, 0.05}], Inset[{0.5, .22, 0}, {.5, .22, 0.05}],
   Inset[{.28, .16, 0}, {.28, .16, -.05}]}}, ImageSize → Medium, Boxed → False]
```

Out[16]=



In 3D the identity will be either

In[17]:=
```
ι3 := TranslationTransform[{0, 0, 0}]
ι := ι3
```

We rotate this 45° around the x-axis

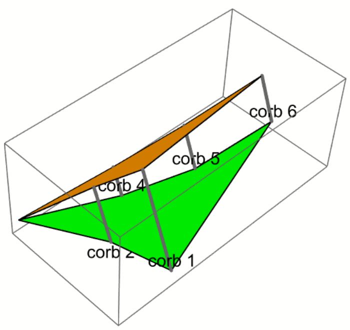
In[19]:=
```
σL1 := RotationTransform[-Pi / 4., {0, 1, 0}]
```

In[20]:=
```
rcorb = σL1@corb
```

Out[20]=
```
{{0.353553, 0.22, 0.353553}, {0.19799, 0.16, 0.19799}, {0., 0., 0.},
 {0.106066, 0.33, 0.106066}, {0.162635, 0.61, 0.162635}, {0.176777, 1., 0.176777}}
```

In[21]:=
```
Graphics3D[{{Green, Polygon[corb]}, {Yellow, Polygon[rcorb]},
  {GrayLevel[.4], Thickness[.01], Line[{corb[[1]], rcorb[[1]]}], Line[{corb[[2]], rcorb[[2]]}],
   Line[{corb[[4]], rcorb[[4]]}], Line[{corb[[5]], rcorb[[5]]}], Line[{corb[[6]], rcorb[[6]]}]},
  {Black, Inset["corb 4", {.15, .33, 0.05}], Inset["corb 5", {.23, .61, .05}],
   Inset["corb 6", {.25, 1, 0.05}], Inset["corb 1", {.5, .22, 0.05}],
   Inset["corb 2", {.28, .16, -.05}]}}, ImageSize → Small]
```

Out[21]=

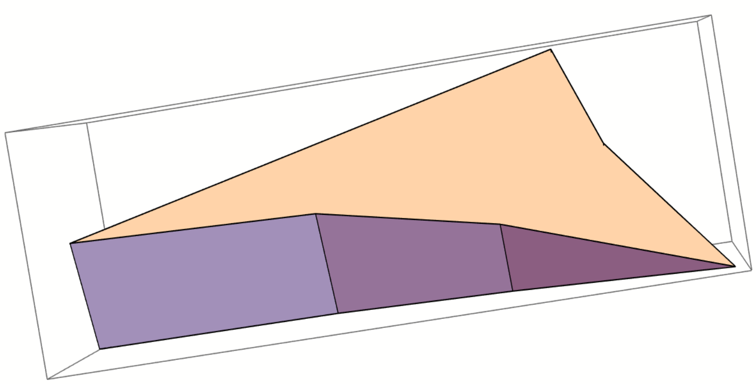


Now connect points of **corb** with points of **rcorb** to get a Polyhedron .

In[22]:=
```
P1 :=
 Polyhedron[{corb, {{0, 0, 0}, corb[[2]], rcorb[[2]]}, {corb[[2]], rcorb[[2]], rcorb[[1]], corb[[1]]},
   {corb[[1]], rcorb[[1]], rcorb[[6]], corb[[6]]}, {corb[[6]], rcorb[[6]], rcorb[[5]], corb[[5]]},
   {corb[[5]], rcorb[[5]], rcorb[[4]], corb[[4]]}, {corb[[4]], rcorb[[4]], {0, 0, 0}},
   rcorb}]
```

In[23]:=
```
Graphics3D[P1]
```

Out[23]=



Since this is important to a physical construction of my lattice we give this again using coordinates

In[24]:=
```
Logo3DA = P1
```

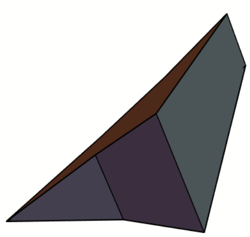
Out[24]=
```
Polyhedron[
 {{{0.5, 0.22, 0}, {0.28, 0.16, 0}, {0, 0, 0}, {0.15, 0.33, 0}, {0.23, 0.61, 0}, {0.25, 1, 0}},
  {{0, 0, 0}, {0.28, 0.16, 0}, {0.19799, 0.16, 0.19799}},
  {{0.28, 0.16, 0}, {0.19799, 0.16, 0.19799}, {0.353553, 0.22, 0.353553}, {0.5, 0.22, 0}},
  {{0.5, 0.22, 0}, {0.353553, 0.22, 0.353553}, {0.176777, 1., 0.176777}, {0.25, 1, 0}},
  {{0.25, 1, 0}, {0.176777, 1., 0.176777}, {0.162635, 0.61, 0.162635}, {0.23, 0.61, 0}},
  {{0.23, 0.61, 0}, {0.162635, 0.61, 0.162635}, {0.106066, 0.33, 0.106066}, {0.15, 0.33, 0}},
  {{0.15, 0.33, 0}, {0.106066, 0.33, 0.106066}, {0, 0, 0}},
  {{0.353553, 0.22, 0.353553}, {0.19799, 0.16, 0.19799}, {0., 0., 0.},
   {0.106066, 0.33, 0.106066}, {0.162635, 0.61, 0.162635}, {0.176777, 1., 0.176777}}}]
```

For our use we make it dynamic

In[25]:=
```
DLP[α_] := Polyhedron[
  α /@ {corb, {{0, 0, 0}, corb[[2]], rcorb[[2]]}, {corb[[2]], rcorb[[2]], rcorb[[1]], corb[[1]]},
    {corb[[1]], rcorb[[1]], rcorb[[6]], corb[[6]]}, {corb[[6]], rcorb[[6]], rcorb[[5]], corb[[5]]},
    {corb[[5]], rcorb[[5]], rcorb[[4]], corb[[4]]}, {corb[[4]], rcorb[[4]], {0, 0, 0}},
    rcorb}]
```

In[26]:=
```
Graphics3D[{GrayLevel[.4], DLP[ι]}, ImageSize → Small, Boxed → False]
```

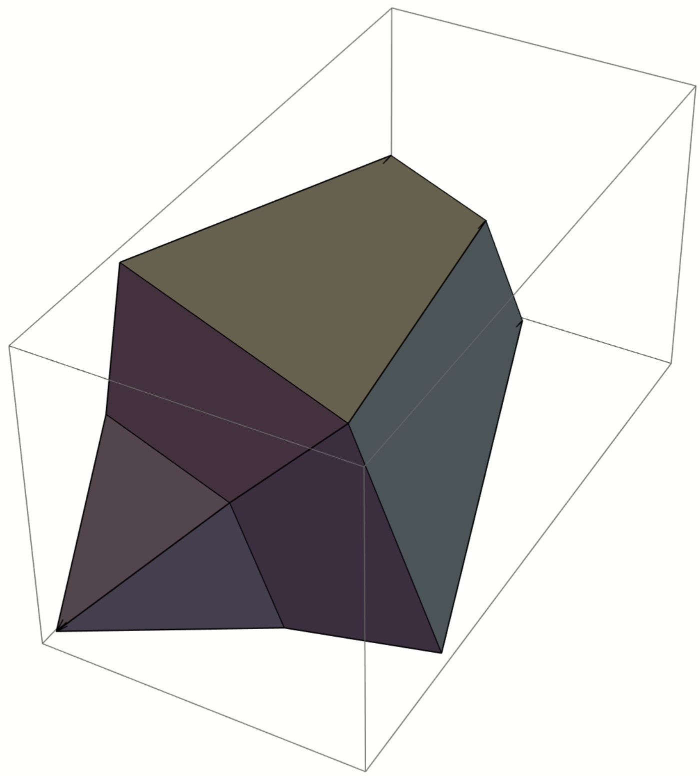
Out[26]=



To this we rotate another 45° to get

In[27]:=
```
Graphics3D[{GrayLevel[.4], DLP[ι], DLP[σL1]}]
```

Out[27]=



We will be using this for corners so we make it dynamic,

In[28]:=
```
Clear[DLC]
```

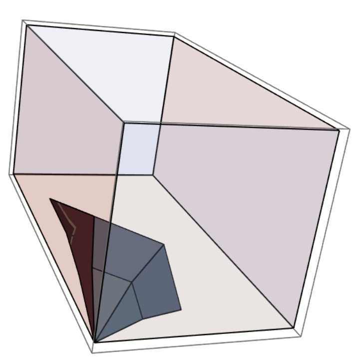
In[29]:=
```
DLC[β_] := {DLP[β], DLP[β@*σL1]}
```

Now we want a fundamental rectangular box

In[30]:=
```
R := {{{0, 0, 0}, {1, 0, 0}, {1, 2, 0}, {0, 2, 0}},
   {{0, 0, 0}, {0, 2, 0}, {0, 2, 1}, {0, 0, 1}}, {{0, 0, 0}, {1, 0, 0}, {1, 0, 1}, {0, 0, 1}},
   {{0, 2, 0}, {0, 2, 1}, {1, 2, 1}, {1, 2, 0}}, {{1, 2, 0}, {1, 2, 1}, {1, 0, 1}, {1, 0, 0}}}
```

In[31]:=
```
Graphics3D[{{GrayLevel[.95], Opacity[.2], Polyhedron[R]}, {GrayLevel[.4], DLC[ι]}},
 ImageSize → Small]
```

Out[31]=



We can partially fill this box with corner pieces to get what I call the Inner Skeleton . We need some more transformations, the first is the the application of σL1 twice, otherwise known as a quarter turn the second inverse or 7 fold composition of σL1

In[32]:=
```
σL2 := σL1@*σL1
σL7 := RotationTransform[Pi / 4, {0, 1, 0}]
```

Then we need some translations and some half turns.

In[34]:=
```
τLx := TranslationTransform[{1, 0, 0}]
τLy := TranslationTransform[{0, 2, 0}]
τLz := TranslationTransform[{0, 0, 1}]
```

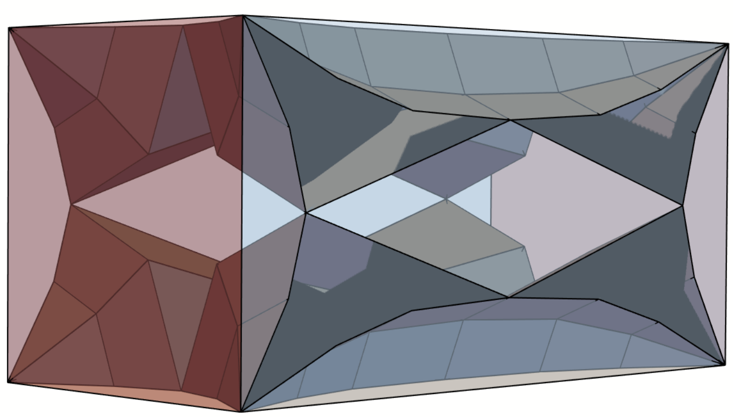
In[37]:=
```
ηL := TransformationFunction[( -1.`  0.`  0.` | 1.`
                                0.` -1.`  0.` | 2.`
                                0.`  0.`  1.` | 0.`
                               ---------------------
                                0.`  0.`  0.` | 1.` )]
```

In[38]:=
```
ηL2 := TransformationFunction[( -1.`  0.`  0.` | 1.`
                                 0.`  1.`  0.` | 0.`
                                 0.`  0.` -1.` | 1.`
                                ---------------------
                                 0.`  0.`  0.` | 1.` )]
```

We now can populate our box

In[39]:=
```
Graphics3D[{{GrayLevel[.9], Opacity[.5], Polyhedron[R]},
  {GrayLevel[.4], DLC[ι], DLC[τLx@*σL2], DLC[ηL], DLC[ηL@*τLx@*σL2], DLC[ηL2],
   DLC[ηL2@*τLx@*σL2], DLC[ηL2@*ηL], DLC[ηL2@*ηL@*τLx@*σL2]}}, Boxed → False]
```

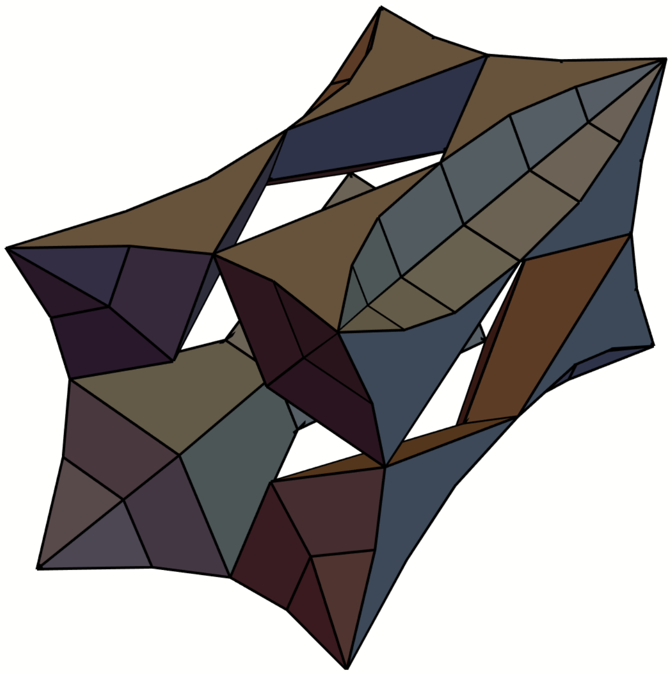
Out[39]=



I call this the Inner Skeleton

In[40]:=
```
LInnerSkeleton := {DLC[ι], DLC[τLx@*σL2], DLC[ηL], DLC[ηL@*τLx@*σL2],
  DLC[ηL2], DLC[ηL2@*τLx@*σL2], DLC[ηL2@*ηL], DLC[ηL2@*ηL@*τLx@*σL2]}
```

In[41]:=
```
Graphics3D[{GrayLevel[.4], LInnerSkeleton}, Boxed → False]
```
Out[41]=



This is not physically stable and more material is needed in the corners to get a rectangular box bounding this. Along the long sides remembering the construction we can write down a polyhedron filling a corner: the sides are

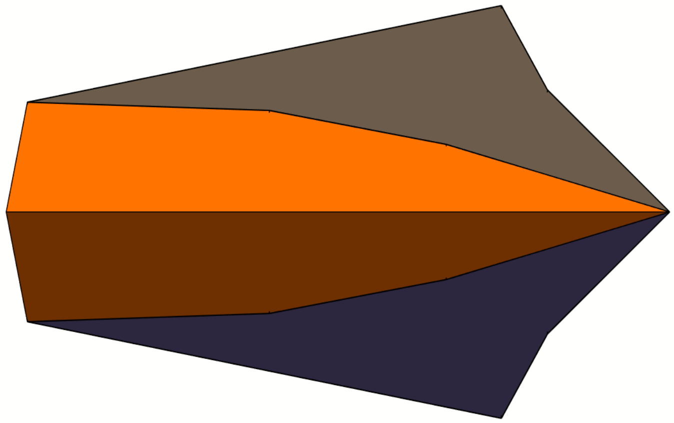
In[42]:=
```
P2 = Chop[{{{0, 0, 0}, corb⟦4⟧, corb⟦5⟧, corb⟦6⟧, {0, 1, 0}},
   {{0, 0, 0}, corb⟦4⟧, corb⟦5⟧, corb⟦6⟧, σL1@corb⟦6⟧, σL1@corb⟦5⟧, σL1@corb⟦4⟧},
   {{0, 0, 0}, σL1@corb⟦4⟧, σL1@corb⟦5⟧, σL1@corb⟦6⟧, σL2@corb⟦6⟧,
    σL2@corb⟦5⟧, σL2@corb⟦4⟧}, {{0, 1, 0}, corb⟦6⟧, σL1@corb⟦6⟧, σL2[corb⟦6⟧]},
   {{0, 0, 0}, σL2@corb⟦4⟧, σL2@corb⟦5⟧, σL2@corb⟦6⟧, {0, 1, 0}}}]
```
Out[42]=
```
{{{0, 0, 0}, {0.15, 0.33, 0}, {0.23, 0.61, 0}, {0.25, 1, 0}, {0, 1, 0}},
 {{0, 0, 0}, {0.15, 0.33, 0}, {0.23, 0.61, 0}, {0.25, 1, 0},
  {0.176777, 1., 0.176777}, {0.162635, 0.61, 0.162635}, {0.106066, 0.33, 0.106066}},
 {{0, 0, 0}, {0.106066, 0.33, 0.106066}, {0.162635, 0.61, 0.162635},
  {0.176777, 1., 0.176777}, {0, 1., 0.25}, {0, 0.61, 0.23}, {0, 0.33, 0.15}},
 {{0, 1, 0}, {0.25, 1, 0}, {0.176777, 1., 0.176777}, {0, 1., 0.25}},
 {{0, 0, 0}, {0, 0.33, 0.15}, {0, 0.61, 0.23}, {0, 1., 0.25}, {0, 1, 0}}}
```

We show the actual coordinates since this is the second piece needing physical construction . For theory we make it dynamic

In[43]:=
```
DLC1[β_] := Polyhedron[β /@ P2]
```

In[44]:=
```
Graphics3D[{{GrayLevel[.4], DLC[ι]}, {Orange, DLC1[ι]}},
 ViewPoint → {Left, Bottom}, Boxed → False]
```

Out[44]=



We also need

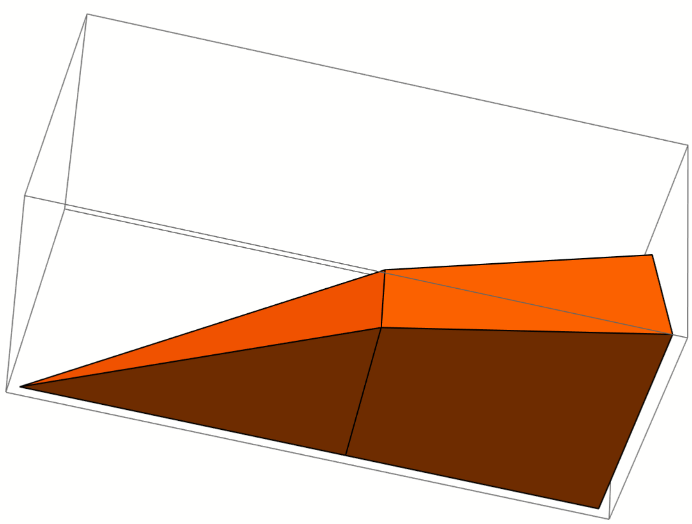
In[45]:=
```
σs1 := TransformationFunction[(1.`  0.`                  0.`                   | 0.`
                               0.`  0.7071067811865476`  -0.7071067811865476`  | 0.`
                               0.`  0.7071067811865476`  0.7071067811865476`   | 0.`
                               ------------------------------------------------------
                               0.`  0.`                  0.`                   | 1.`)];
```

In[46]:=
```
σs7 := TransformationFunction[Inverse[TransformationMatrix[σs1]]]
```

In[47]:=
```
P3 := {{{0, 0, 0}, {0.28`, 0.16`, 0}, {0.5`, 0.22`, 0}, {0.5`, 0, 0}},
   {{0.5`, 0.22`, 0}, {0.5`, 0.1555`, 0.15556349186104046`}, {0.5`, 0, 0.22`}, {0.5`, 0, 0}},
   {{0, 0, 0}, {0.28`, 0, 0}, {0.28`, 0, 0.16`}},
   {{0.28`, 0, 0}, {0.5`, 0, 0}, {0.5`, 0, 0.22`}, {0.28`, 0, 0.16`}},
   {{0, 0, 0}, {0.28`, 0.16`, 0}, {0.28`, 0.11314`, 0.11314`}}, {{0.28`, 0.16`, 0},
    {0.5`, 0.22`, 0}, {0.5`, 0.1556`, 0.1556`}, {0.28`, 0.11314`, 0.11314`}},
   {{0, 0, 0}, {0.28`, 0.11314`, 0.113134`}, {0.28`, 0, 0.16`}}, {{0.28`, 0.11314`, 0.11314`},
    {0.28`, 0, 0.16`}, {0.5`, 0, 0.22`}, {0.5`, 0.1556`, 0.1556`}}}
```

In[48]:=
```
Graphics3D[{Orange, Polyhedron[P3]}]
```

Out[48]=



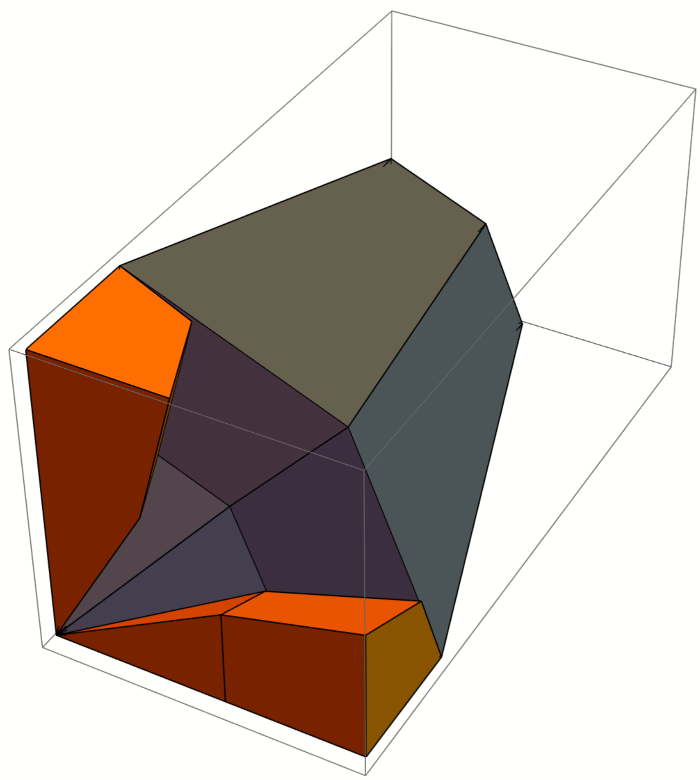
```
In[49]:= DLC2[γ_] := Polyhedron[γ /@ P3]
```

Next we combine in a corner

```
In[50]:= FullCorner[α_] :=
  {{GrayLevel[.4], DLC[α]}, {Orange, DLC1[α], DLC2[α], DLC2[α@*σL2@*σS7@*σS7]}}
```

```
In[51]:= Graphics3D[FullCorner[ι]]
```

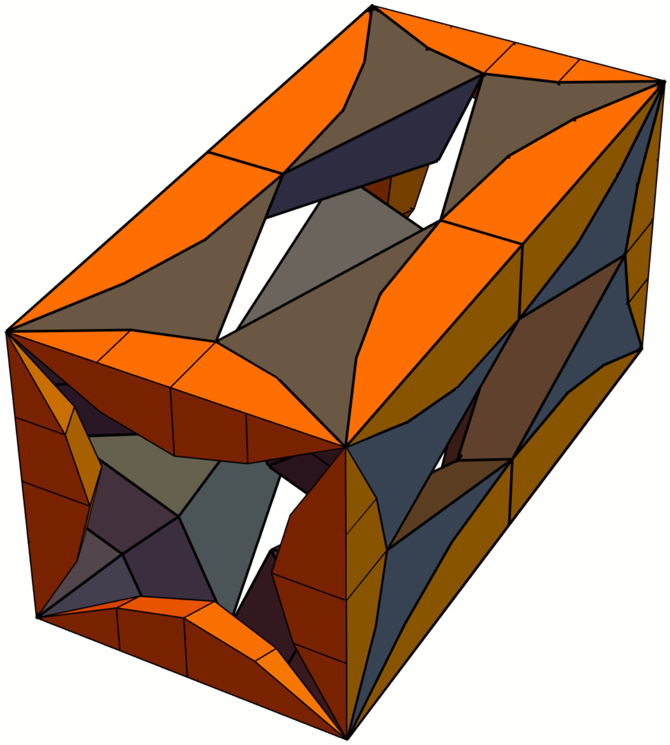
Out[51]=



```
In[52]:= LBrick[β_] := Graphics3D[{FullCorner[β], FullCorner[β@*τLx@*σL2], FullCorner[β@*ηL],
    FullCorner[β@*ηL@*τLx@*σL2], FullCorner[β@*ηL2], FullCorner[β@*ηL2@*τLx@*σL2],
    FullCorner[β@*ηL2@*ηL], FullCorner[β@*ηL2@*ηL@*τLx@*σL2]}, Boxed → False]
```

In[53]:=
```
LBrick[ι]
```

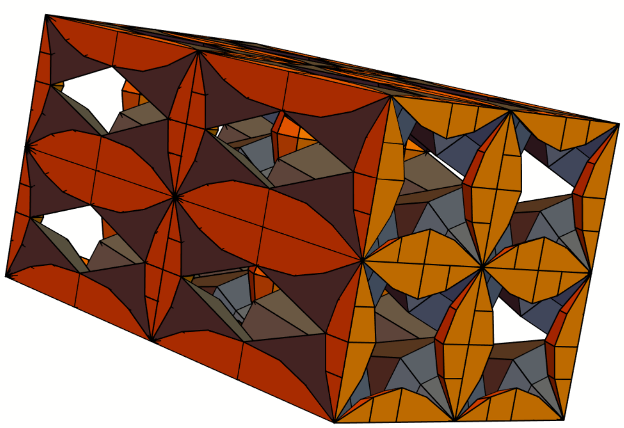
Out[53]=



This will now be our building block

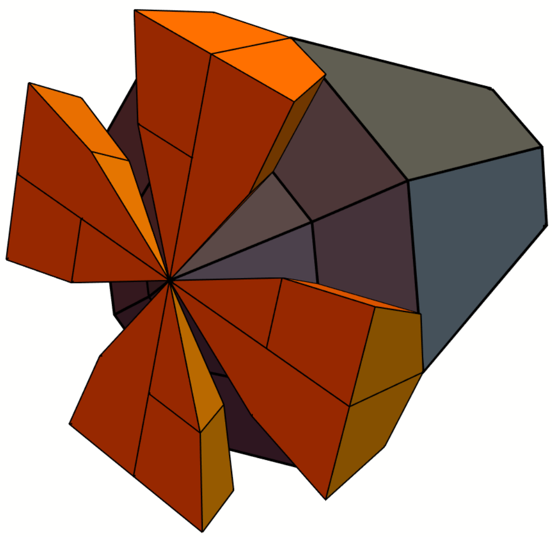
In[54]:=
```
Show[LBrick[ι], LBrick[τLy], LBrick[τLx], LBrick[τLx@*τLy],
 LBrick[τLz@*τLx], LBrick[τLz@*τLx@*τLy], LBrick[τLz], LBrick[τLz@*τLy]]
```

Out[54]=



In[56]:=
```
Graphics3D[{FullCorner[ι], FullCorner[σL2],
  FullCorner[σL2@⋆σL2], FullCorner[σL2@⋆σL2@⋆σL2]}, Boxed → False]
```

Out[56]=

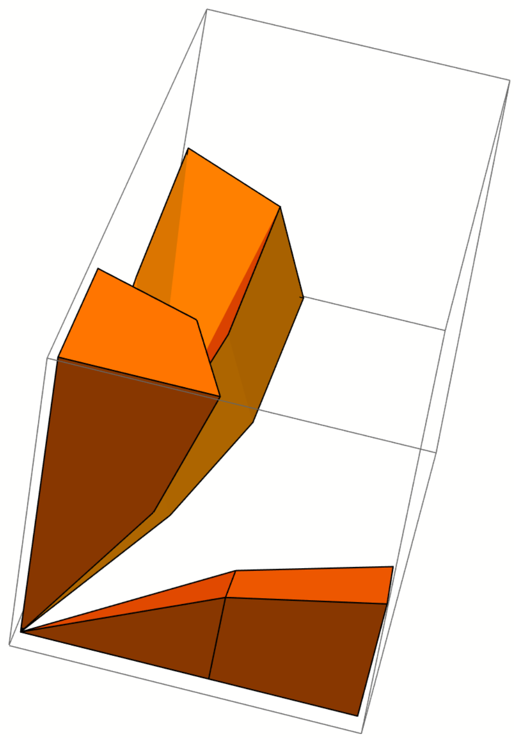


Outer corner of brick

In[59]:=
```
OCB[α_] := {Orange, DLC1[α], DLC2[α], DLC2[α@⋆σL2@⋆σs7@⋆σs7]}
```

In[60]:= `Graphics3D[OCB[ι]]`

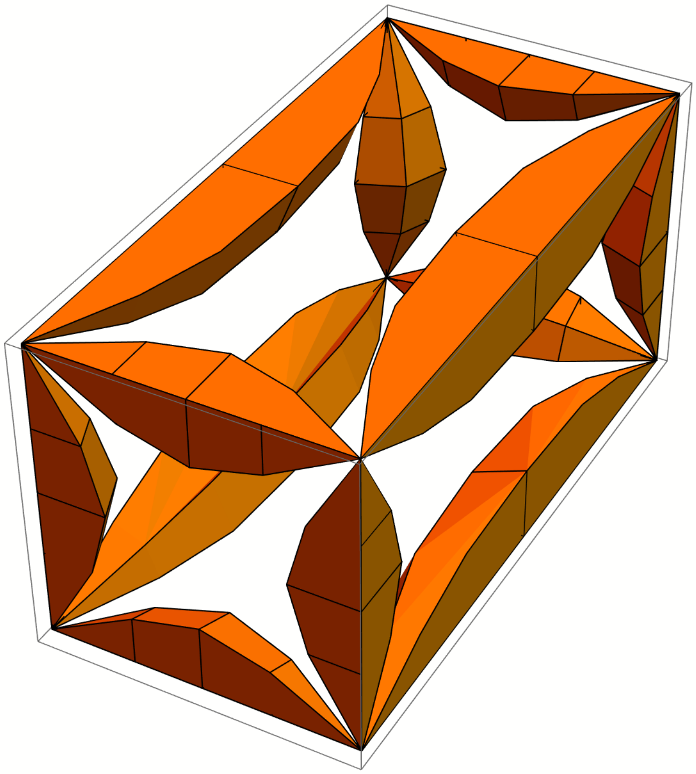
Out[60]=



In[64]:=
```
OuterSkelBrick[β_] :=
 Graphics3D[{OCB[β], OCB[β@⋆τLx@⋆σL2], OCB[β@⋆ηL], OCB[β@⋆ηL@⋆τLx@⋆σL2],
   OCB[β@⋆ηL2], OCB[β@⋆ηL2@⋆τLx@⋆σL2], OCB[β@⋆ηL2@⋆ηL], OCB[β@⋆ηL2@⋆ηL@⋆τLx@⋆σL2]}]
```

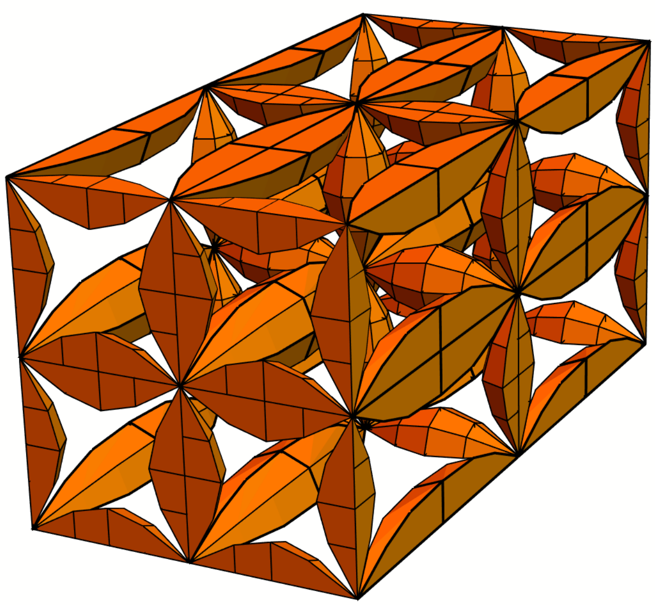
In[65]:=
```
OuterSkelBrick[ι]
```

Out[65]=



In[68]:=
```
OuterSkel2by2 = Show[OuterSkelBrick[ι], OuterSkelBrick[τLy], OuterSkelBrick[τLx],
  OuterSkelBrick[τLx@*τLy], OuterSkelBrick[τLz@*τLx], OuterSkelBrick[τLz@*τLx@*τLy],
  OuterSkelBrick[τLz], OuterSkelBrick[τLz@*τLy], Boxed → False]
```

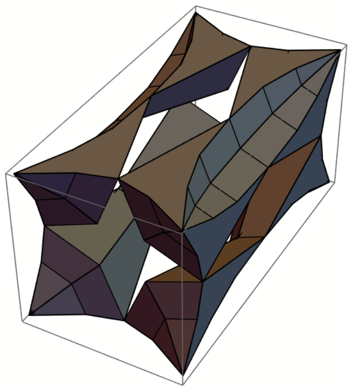
Out[68]=



We now work on the inner skeleton, first making it dynamic

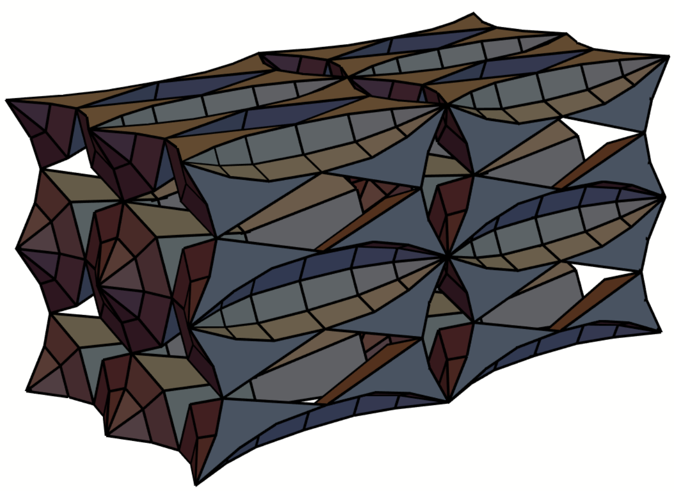
```
In[75]:= DInSkelBrick[γ_] := {DLC[γ], DLC[γ@* τLx@* σL2], DLC[γ@* ηL], DLC[γ@* ηL@* τLx@* σL2],
           DLC[γ@* ηL2], DLC[γ@* ηL2@* τLx@* σL2], DLC[γ@* ηL2@* ηL], DLC[γ@* ηL2@* ηL@* τLx@* σL2]}

In[73]:= Graphics3D[{GrayLevel[.4], DInSkel[ι]}, ImageSize → Small]
Out[73]=
```



Now in the 2 by 2

```
In[82]:= InnerSkel2by2 =
          Graphics3D[{GrayLevel[.4], DInSkelBrick[ι], DInSkelBrick[τLy], DInSkelBrick[τLx],
            DInSkelBrick[τLx@* τLy], DInSkelBrick[τLz@* τLx], DInSkelBrick[τLz@* τLx@* τLy],
            DInSkelBrick[τLz], DInSkelBrick[τLz@* τLy]}, Boxed → False]
Out[82]=
```



Notice that neither the outer nor inner skeleton is physically stable, but the combination is .