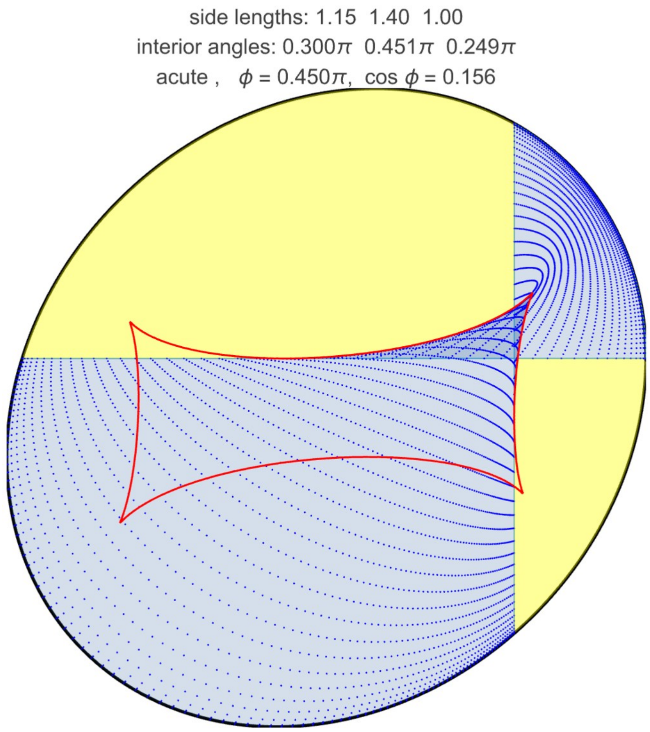
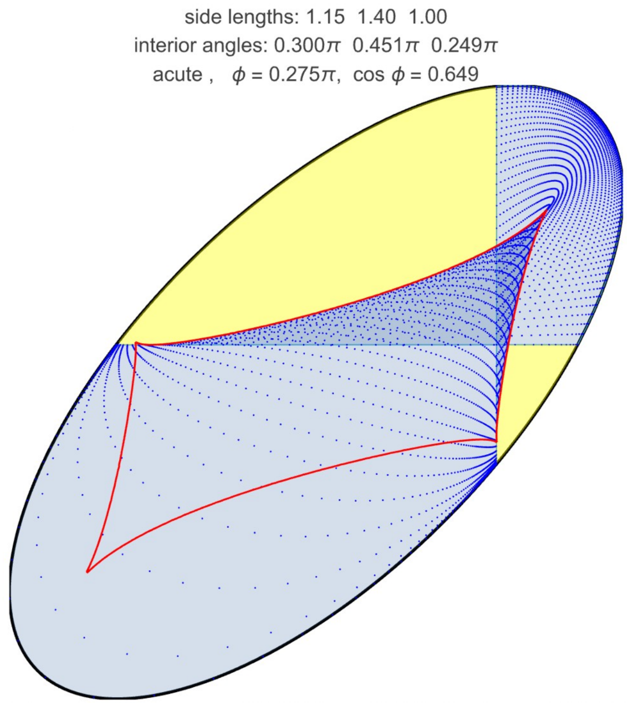
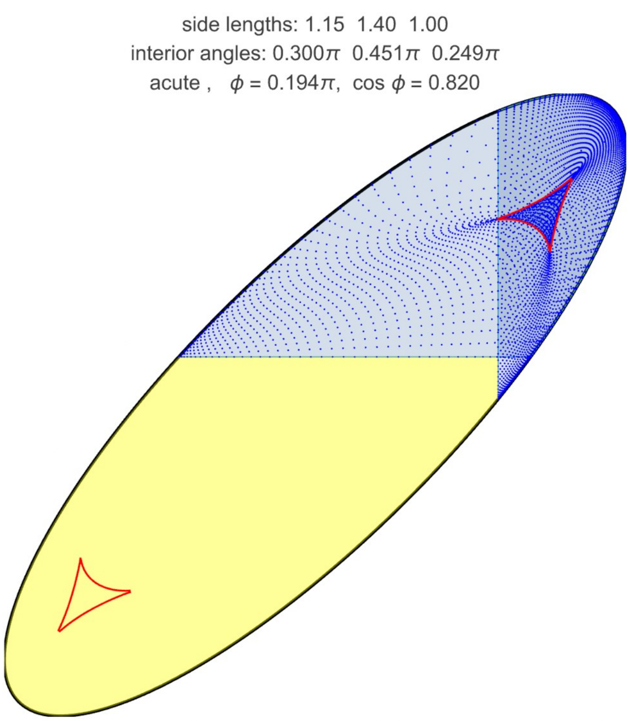
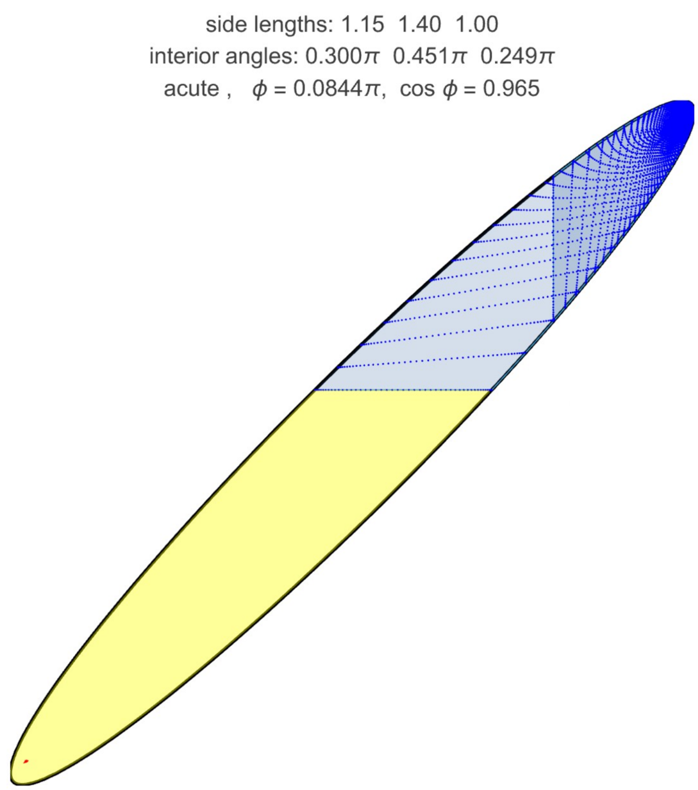
Several examples of slices of our solid (blue) in “cosines space.” (See the “4 Problems” document for more explanation. Remember that φ here really means γ, which is fixed for each slice.)

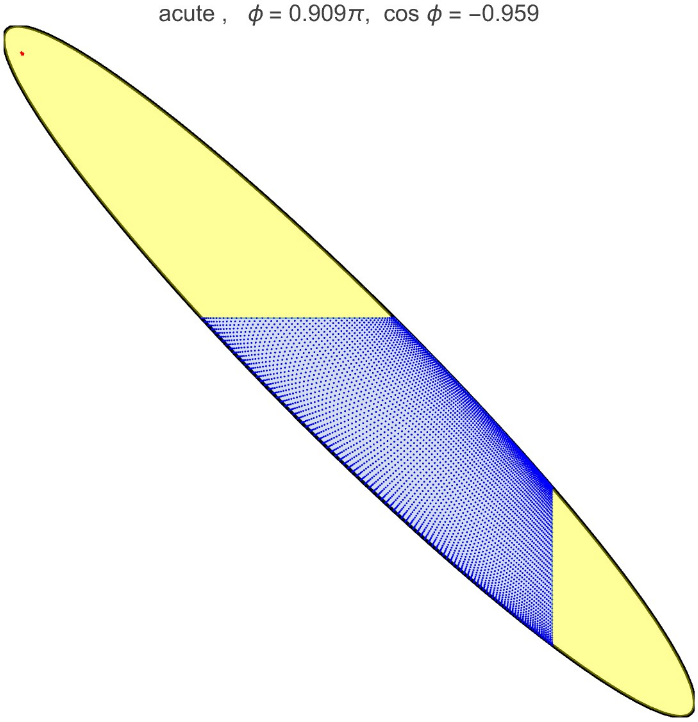




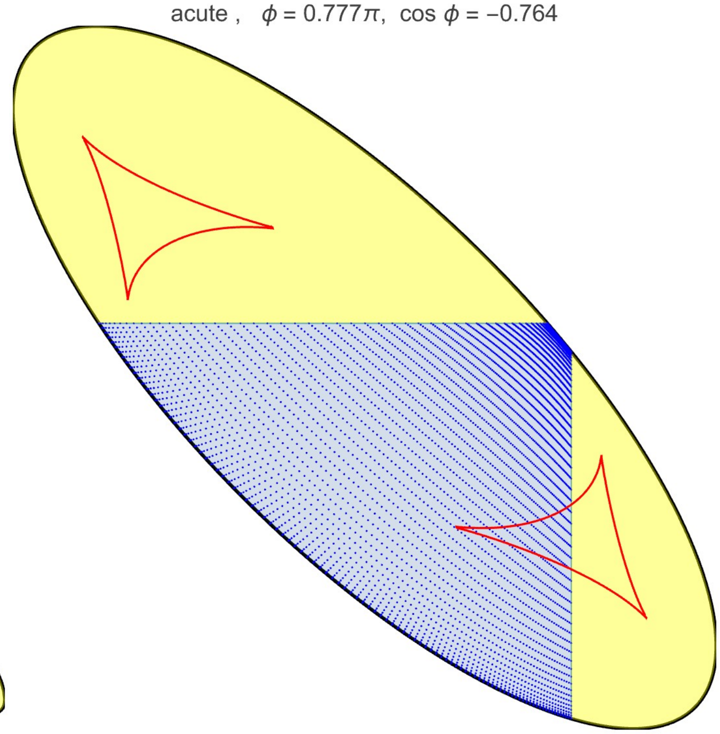




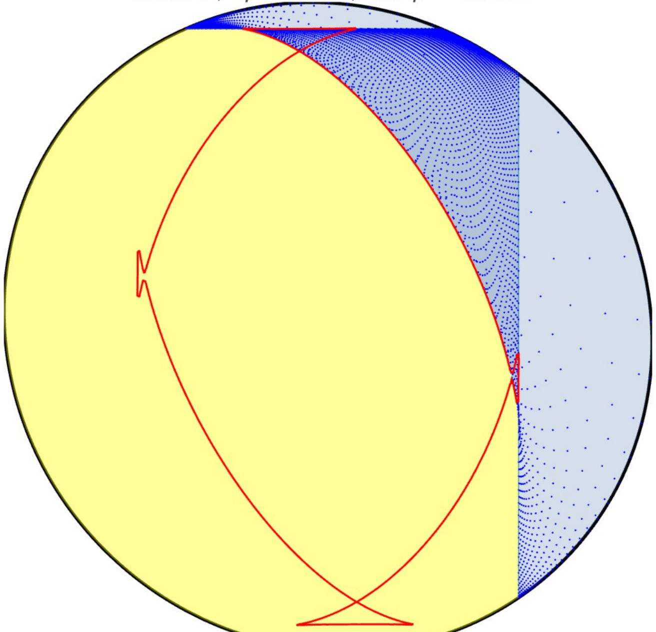
side lengths: 1.15 1.40 1.00
interior angles: 0.300π 0.451π 0.249π
acute , $\phi = 0.909\pi$, $\cos\phi = -0.959$



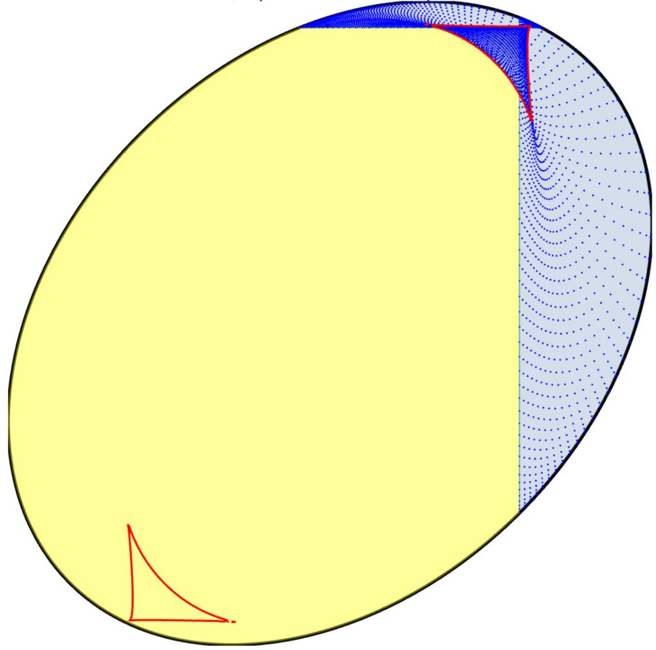
side lengths: 1.15 1.40 1.00
interior angles: 0.300π 0.451π 0.249π
acute , $\phi = 0.777\pi$, $\cos\phi = -0.764$



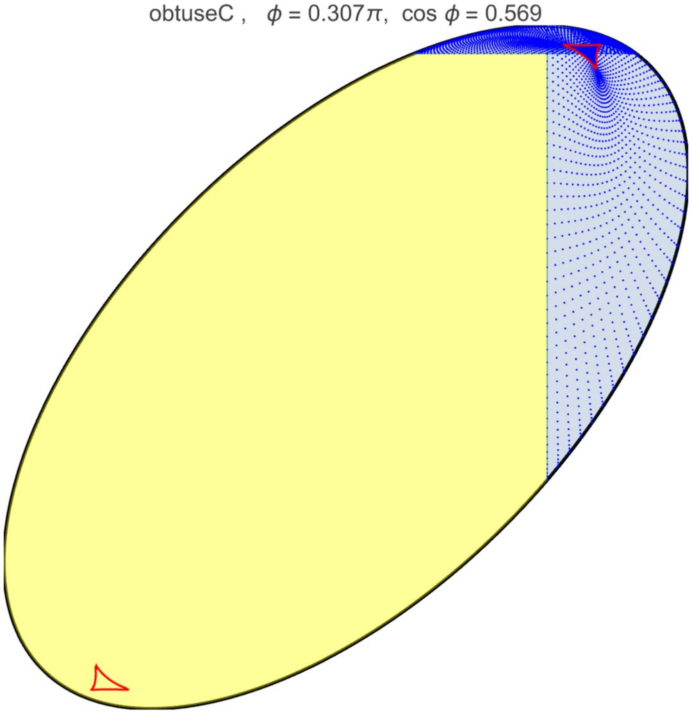
side lengths: 2.87 1.40 3.46
interior angles: 0.300π 0.129π 0.571π
obtuseC , $\phi = 0.516\pi$, $\cos\phi = -0.0492$



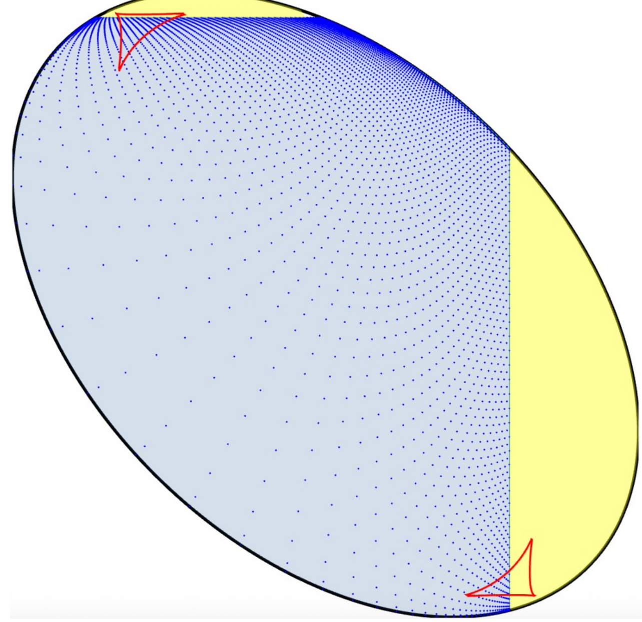
side lengths: 2.87 1.40 3.46
interior angles: 0.300π 0.129π 0.571π
obtuseC , $\phi = 0.401\pi$, $\cos\phi = 0.306$



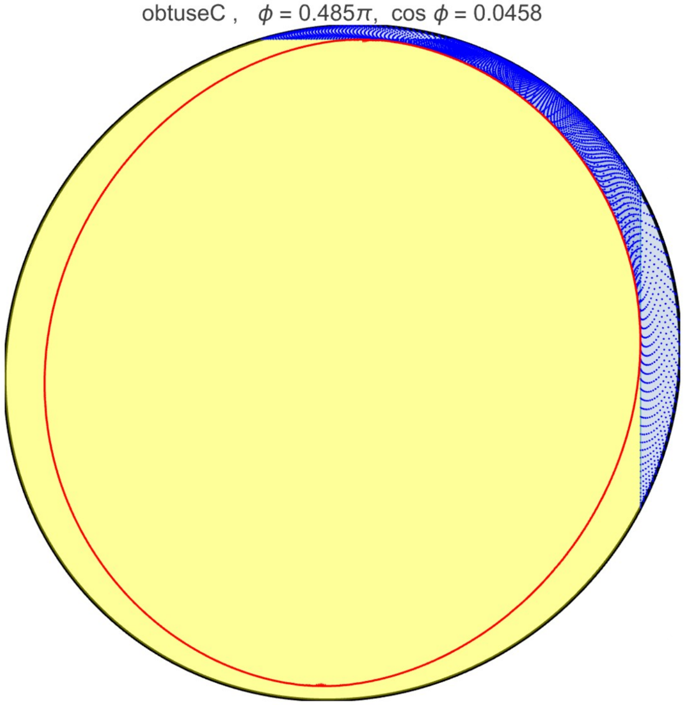
side lengths: 2.87 1.40 3.46
interior angles: 0.300π 0.129π 0.571π
obtuseC , $\phi = 0.307\pi$, $\cos\phi = 0.569$



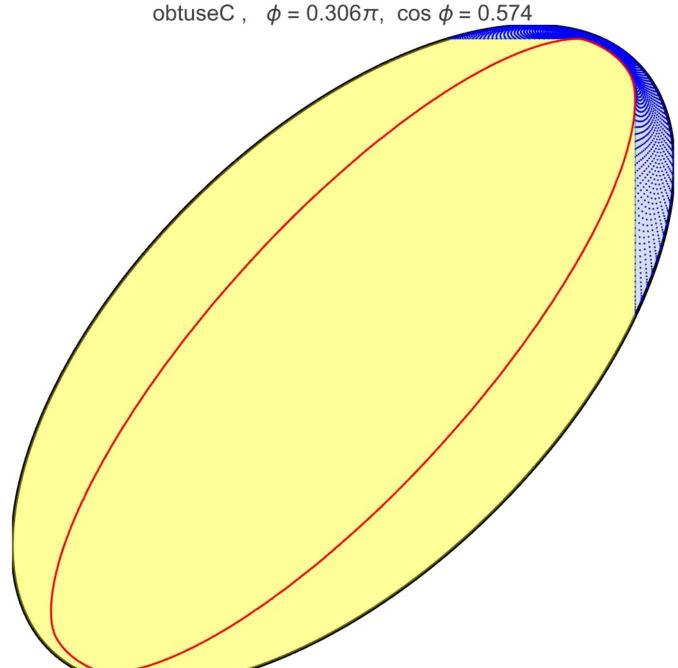
side lengths: 2.87 1.40 3.46
interior angles: 0.300π 0.129π 0.571π
obtuseC , $\phi = 0.633\pi$, $\cos\phi = -0.407$



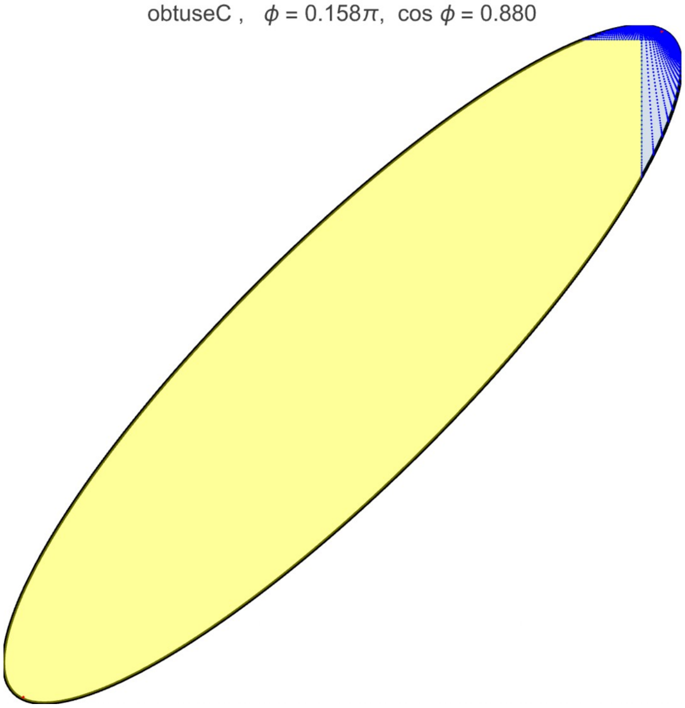
side lengths: 2.32 1.40 3.46
interior angles: 0.156π 0.0917π 0.752π
obtuseC , $\phi = 0.485\pi$, $\cos\phi = 0.0458$



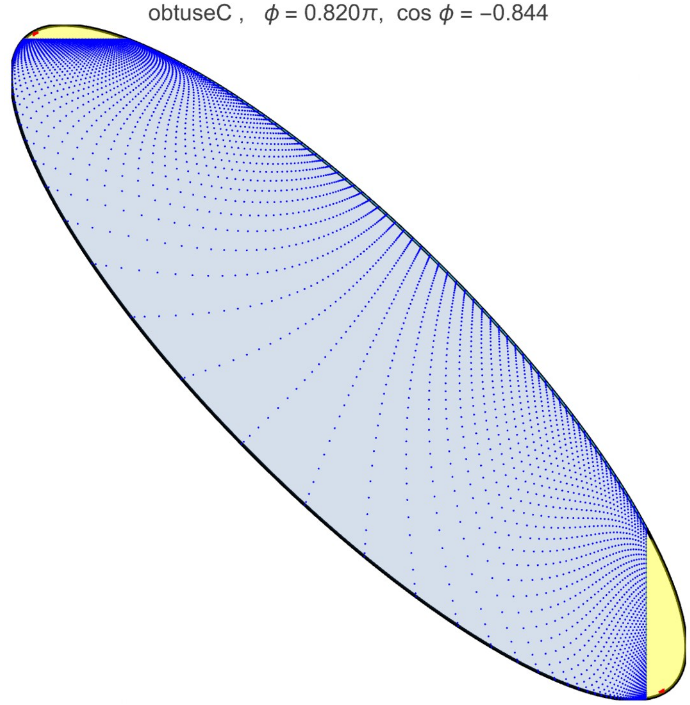
side lengths: 2.32 1.40 3.46
interior angles: 0.156π 0.0917π 0.752π
obtuseC , $\phi = 0.306\pi$, $\cos\phi = 0.574$



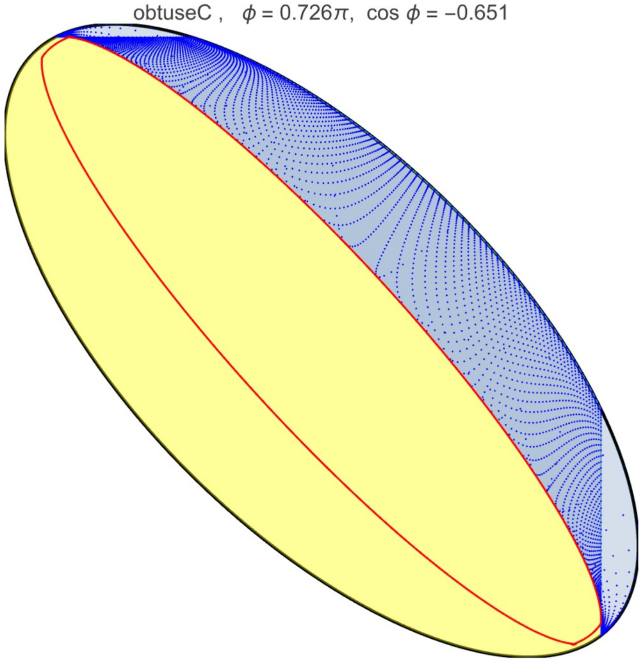
side lengths: 2.32 1.40 3.46
interior angles: 0.156π 0.0917π 0.752π
obtuseC , ϕ = 0.158π, cos ϕ = 0.880



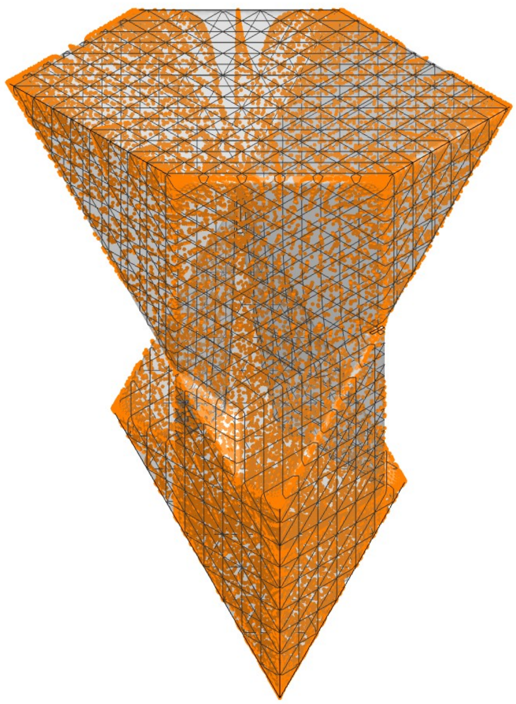
side lengths: 2.32 1.40 3.46
interior angles: 0.156π 0.0917π 0.752π
obtuseC , ϕ = 0.820π, cos ϕ = −0.844

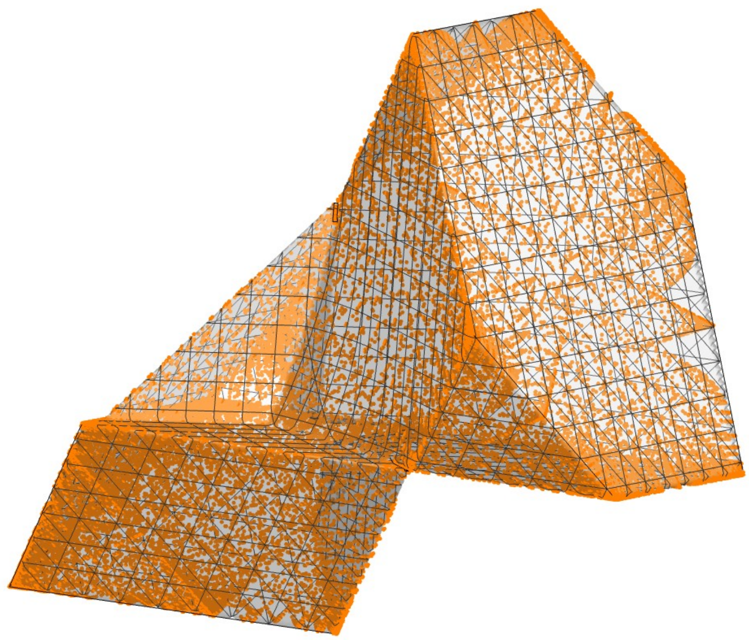


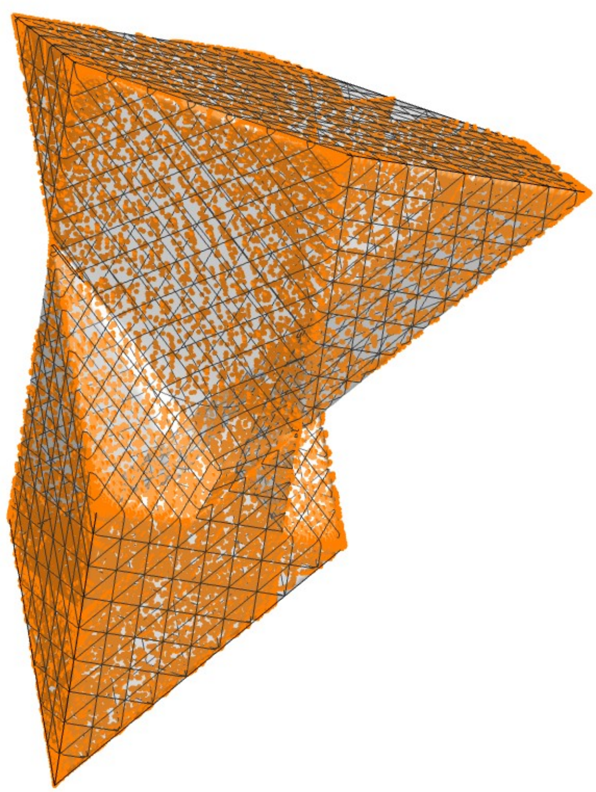
side lengths: 2.32 1.40 3.46
interior angles: 0.156π 0.0917π 0.752π
obtuseC , ϕ = 0.726π, cos ϕ = −0.651

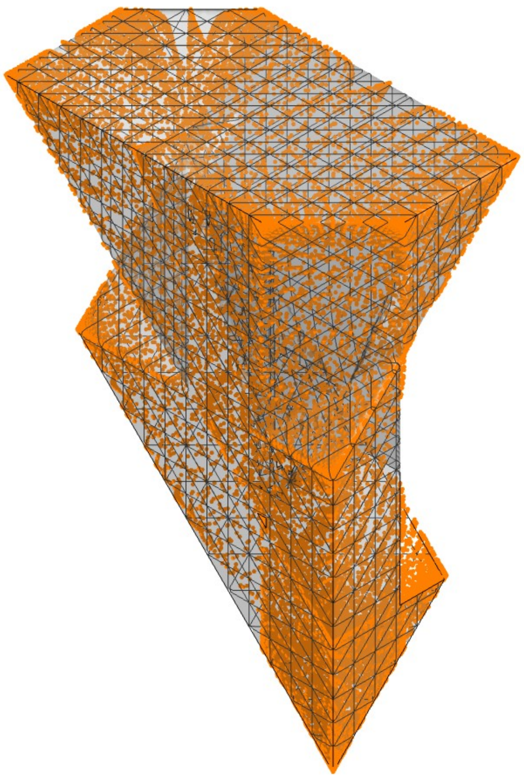


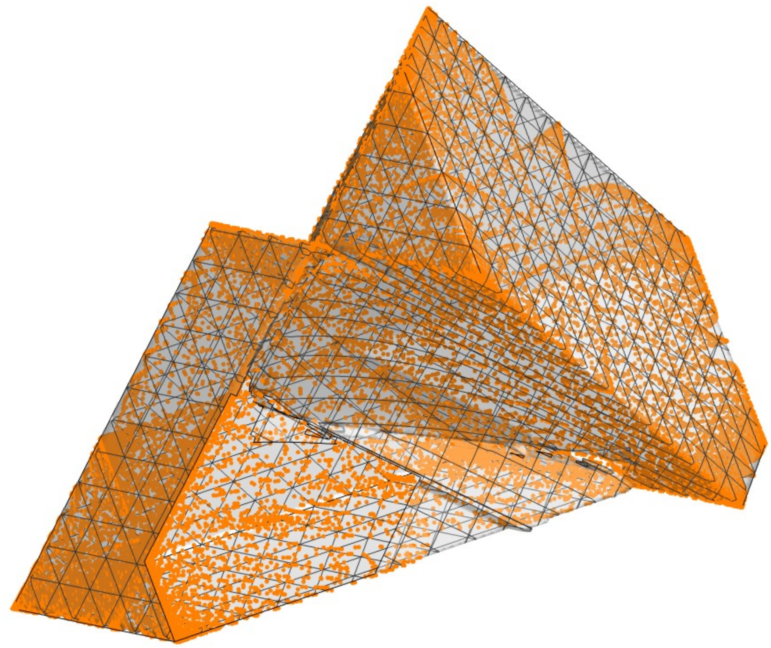
Here are some examples of our solid in “angles space.” The first three are different views of the same solid for an acute triangle case. The last three are different views of the same solid for an obtuse triangle case.

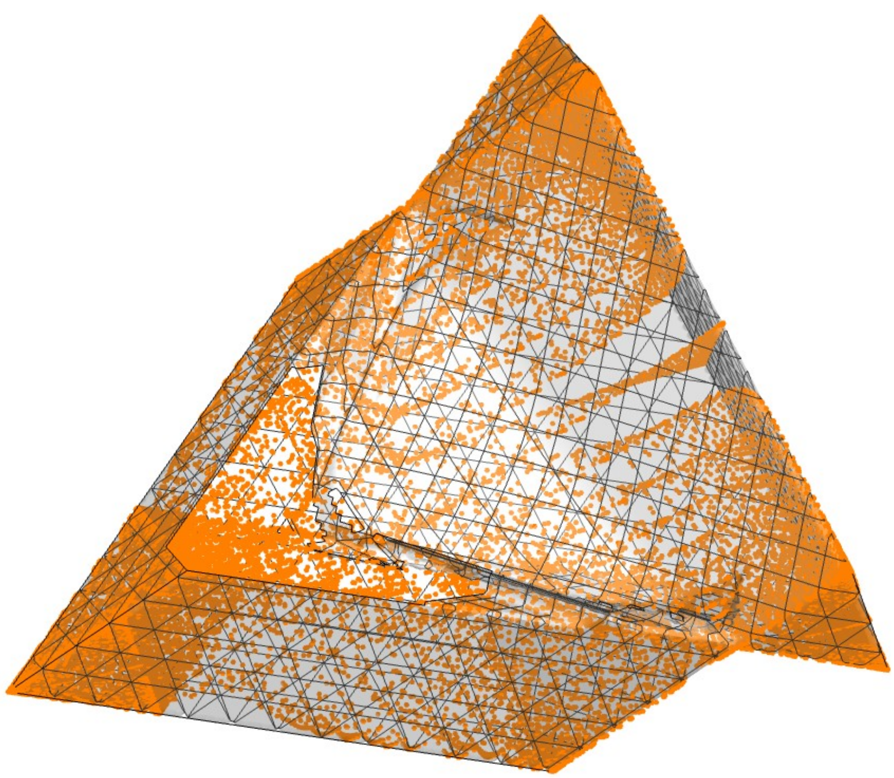












Here is some of the Mathematica code used to create the boundary (gray) for the generated data points of the solid (orange).

```
spike = RegionPlot3D[
     (* The basic spherical geometry rules ("tetrahedral rules") *)
     (     α + β + γ ≤ 2 π
       &&  α ≤ β + γ
       &&  β ≤ γ + α
       &&  γ ≤ α + β  )
   &&  (* Additional spherical geometry rules for acute triangle case *)
   ( obtuse || (
          AA + β + γ ≤ 2 π
      &&  α + BB + γ ≤ 2 π
      &&  α + β + CC ≤ 2 π
     (*
      &&  β + γ - α  <= 2(BB+CC)
      &&  γ + α - β  <= 2(CC+AA)
      &&  α + β - γ  <= 2(AA+BB)
     *)
    ) )
   &&  (* Simple cosines rules for acute triangle case *)
   ( obtuse || (
         ( α ≥ AA  ||  Cos[CC] Cos[β] + Cos[BB] Cos[γ]  > 0 ) &&
         ( β ≥ BB  ||  Cos[AA] Cos[γ] + Cos[CC] Cos[α]  > 0 )  &&
         ( γ ≥ CC  ||  Cos[BB] Cos[α] + Cos[AA] Cos[β]  > 0 ) ) )
   && (* Grunert system discriminant rule for acute triangle case *)
     ( obtuse || (
       disc[x1, x2, x3, y1, y2, y3, Cos[α], Cos[β], Cos[γ]] ≥ 0   ||
          ( ( α ≥ AA || β < BB || γ < CC ) && ( α < AA || β ≥ BB || γ < CC ) && ( α < AA || β < BB || γ ≥  CC ) ) ) )
   && (* LO condition for obtuse triangle case *)
   ( Not[obtuse] || (
      ( disc[x1, x2, x3, y1, y2, y3, Cos[α], Cos[β], Cos[γ]] < 0 ) ||
       ( ( q[x1, x2, y1, y2, BB, CC, Cos[β], Cos[γ]] > 0 )
         && ( 2 Cos[β] ^2 ≥ 1 + Cos[2 BB]  ||  (1 + Cos[2 AA]) Cos[γ] ^2 ≥ Cos[2 AA] + Cos[2 CC] )
         && ( 2 Cos[γ] ^2 ≥ 1 + Cos[2 CC]  ||  (1 + Cos[2 AA]) Cos[β] ^2 ≥ Cos[2 AA] + Cos[2 BB] ) )
    ) )
   && (* RO condition for obtuse triangle case *)
   ( Not[obtuse] || ro[ϕ1, ϕ2, ϕ3, AA, BB, CC, Cos[α], Cos[β], Cos[γ]] )
```

```
disc[x1_, x2_, x3_, y1_, y2_, y3_, c1_, c2_, c3_] := Module[ {L, R, H},
  Evaluate[ N[4 H (H - 2 R)^3 - (L^2 + R^2 - 10 H R - 2 H^2)^2]  /. {
    L → 2 ((-1 + c1^2) (1 - x1) y1 + (-1 + c2^2) (1 - x2) y2 + (-1 + c3^2) (1 - x3) y3 + (1 - c1 c2 c3) (y1 + y2 + y3)),
    R → 2 ((-1 + c1^2) x1 (1 + x1) + (-1 + c2^2) x2 (1 + x2) + (-1 + c3^2) x3 (1 + x3) + (1 - c1 c2 c3) (1 + x1 + x2 + x3)),
    H →  1 - c1^2 - c2^2 + 2 c1 c2 c3 - c3^2 } ]];

q[x2_, y2_, x3_, y3_, B_, C_, c2_, c3_] := Module[{ j2, k2, m2, j3, k3, m3 },
  Evaluate[ N[(j2 m3 - j3 m2)^2 + (k2 m3 - k3 m2)^2 - (j2 k3 - j3 k2)^2]  /. {
    j2 → (1 - c2^2) (1 - c3^2) x2 , k2 → (1 - c2^2) (1 - c3^2) y2 , m2 → (1 - c3^2) (c2^2 - Cos[2 B]),
    j3 → (1 - c2^2) (1 - c3^2) x3 , k3 → (1 - c2^2) (1 - c3^2) y3 , m3 → (1 - c2^2) (c3^2 - Cos[2 C]) } ]];

ro[ϕϕ1_, ϕϕ2_, ϕϕ3_, AA_, BB_, CC_, cc1_, cc2_, cc3_] := Module[{ϕ1, ϕ2, ϕ3, c1, c2, c3, C1,
    cA, cB, cC, DD, x1, x2, x3, y1, y2, y3, c2cap, c3cap, c2cup, c3cup, answer},
   ϕ1 = N[ϕϕ1]; ϕ2 = N[ϕϕ2]; ϕ3 = N[ϕϕ3];  c1 = N[cc1]; c2 = N[cc2]; c3 = N[cc3];
   C1 = c1^2;  cA = N[Cos[AA]];  cB = N[Cos[BB]]; N[cC = Cos[CC]];
   x1 = Cos[ϕ1]; x2 = Cos[ϕ2]; x3 = Cos[ϕ3]; y1 = Sin[ϕ1]; y2 = Sin[ϕ2]; y3 = Sin[ϕ3];
   c2cap  = Sign[c1] Sqrt[((Cos[ϕ1 - ϕ2] - Cos[ϕ1 - ϕ3]) C1 + Cos[ϕ1 - ϕ3] - Cos[ϕ1 - ϕ2] Cos[ϕ2 - ϕ3] +
         Abs[Sin[(ϕ2 - ϕ3) / 2]] Sin[ϕ1 - ϕ2] Sqrt[2 (1 + Cos[ϕ2 - ϕ3] - 2 C1)]) /
      (1 - C1 + Cos[ϕ1 - ϕ2] C1 - Cos[ϕ1 - ϕ2] Cos[ϕ2 - ϕ3]  + Abs[Sin[(ϕ2 - ϕ3) / 2]] Sin[ϕ1 - ϕ2] Sqrt[2 (1 + Cos[ϕ2 - ϕ3] - 2 C1)])];
   c3cap  = Sqrt[((Cos[ϕ1 - ϕ3] - Cos[ϕ1 - ϕ2]) C1 + Cos[ϕ1 - ϕ2] - Cos[ϕ1 - ϕ3] Cos[ϕ2 - ϕ3] +
        Abs[Sin[(ϕ2 - ϕ3) / 2]] Sin[ϕ1 - ϕ3] Sqrt[2 (1 + Cos[ϕ2 - ϕ3] - 2 C1)]) /
     (1 - C1 + Cos[ϕ1 - ϕ3] C1 - Cos[ϕ1 - ϕ3] Cos[ϕ2 - ϕ3]  + Abs[Sin[(ϕ2 - ϕ3) / 2]] Sin[ϕ1 - ϕ3] Sqrt[2 (1 + Cos[ϕ2 - ϕ3] - 2 C1)])];
   c2cup  = Sqrt[((Cos[ϕ1 - ϕ2] - Cos[ϕ1 - ϕ3]) C1 + Cos[ϕ1 - ϕ3] - Cos[ϕ1 - ϕ2] Cos[ϕ2 - ϕ3] -
        Abs[Sin[(ϕ2 - ϕ3) / 2]] Sin[ϕ1 - ϕ2] Sqrt[2 (1 + Cos[ϕ2 - ϕ3] - 2 C1)]) /
     (1 - C1 + Cos[ϕ1 - ϕ2] C1 - Cos[ϕ1 - ϕ2] Cos[ϕ2 - ϕ3]  - Abs[Sin[(ϕ2 - ϕ3) / 2]] Sin[ϕ1 - ϕ2] Sqrt[2 (1 + Cos[ϕ2 - ϕ3] - 2 C1)])];
   c3cup  = Sign[c1] Sqrt[((Cos[ϕ1 - ϕ3] - Cos[ϕ1 - ϕ2]) C1 + Cos[ϕ1 - ϕ2] - Cos[ϕ1 - ϕ3] Cos[ϕ2 - ϕ3] -
         Abs[Sin[(ϕ2 - ϕ3) / 2]] Sin[ϕ1 - ϕ3] Sqrt[2 (1 + Cos[ϕ2 - ϕ3] - 2 C1)]) /
      (1 - C1 + Cos[ϕ1 - ϕ3] C1 - Cos[ϕ1 - ϕ3] Cos[ϕ2 - ϕ3]  - Abs[Sin[(ϕ2 - ϕ3) / 2]] Sin[ϕ1 - ϕ3] Sqrt[2 (1 + Cos[ϕ2 - ϕ3] - 2 C1)])];
   DD  = disc[x1, x2, x3, y1, y2, y3, c1, c2, c3];
   answer = True;
   If[ Abs[c1]  ≤ Abs[cA],
    If[ Abs[Im[c2cap]] < .0001 && Abs[Im[c3cap]] < .0001 && Abs[Im[c2cup]] < .0001 && Abs[Im[c3cup]] < .0001 ,
     c2cap  = Re[c2cap];  c3cap  = Re[c3cap];  c2cup  = Re[c2cup];  c3cup  = Re[c3cup];
      answer = ((cC c2 + cB c3  ≥ 0) && (c2 ≥ c2cap  || c3 ≥ c3cap) &&
        (c2 ≥ c2cup  || c3 ≥ c3cup && (c2 ≥ c2cap || c3 ≥ cC || DD ≥ 0)  &&
           (c3 ≥ c3cup || c2 ≥ cB || DD ≥ 0))) ]];
   If[c1 < cA,  answer = c2 ≤ cB  && c3 ≤ cC];
   If[c1 > -cA,  answer = (c2 ≥ 0 || c3 ≥ cC) && (c3 ≥ 0 || c2 ≥ cB) && (c2 ≥ cB || c3 ≥ cC || D < 0)];
   answer
  ];
```

The following function is used to convert the “tilt angles” (dihedral angles involving the base plane, i.e. the plane containing the fixed triangle), which are varied independently, to the “view angles” (angles at the varying vertex of the tetrahedron). This function is used to generate the data points for the “spikes” (the orange points in the orange and gray figures). This method for generating the data points is *completely different* than the “extended Sullivan method” used to generate the data points (blue) for the “slices” seen in the blue and yellow figures.

```
tiltToViewAngles[A_, B_, C_, τ1_, τ2_, τ3_] := Module[
   (*
      A, B, C are interior angles of the case triangle;
      τ1, τ2, τ3 are "tilt angles" (dihedral angles involving the base);
      δ1, δ2, δ3 are the other dihedral angles;
      α, β, γ are the "view angles" (at optical center)
   *)
   {δ1, δ2, δ3, α, β, γ, sτ1, sτ2, sτ3, cτ1, cτ2, cτ3, sδ1, sδ2, sδ3, cδ1, cδ2, cδ3},
    sτ1 = Sin[N[τ1]]; sτ2 = Sin[N[τ2]]; sτ3 = Sin[N[τ3]];
    cτ1 = Cos[N[τ1]]; cτ2 = Cos[N[τ2]]; cτ3 = Cos[N[τ3]];
    cδ1 = sτ2 sτ3 Cos[N[A]] - cτ2 cτ3;
    cδ2 = sτ3 sτ1 Cos[N[B]] - cτ3 cτ1;
    cδ3 = sτ1 sτ2 Cos[N[C]] - cτ1 cτ2;
    sδ1 = Sqrt[1 - cδ1^2]; sδ2 = Sqrt[1 - cδ2^2]; sδ3 = Sqrt[1 - cδ3^2];
    α = ArcCos[ (cδ1 + cδ2 cδ3) / (sδ2 sδ3)];
    β = ArcCos[ (cδ2 + cδ3 cδ1) / (sδ3 sδ1)];
    γ = ArcCos[ (cδ3 + cδ1 cδ2) / (sδ1 sδ2)];
   If [ Abs[Im[α]] > .001 || Abs[Im[β]] > .001 || Abs[Im[γ]] > .001 ,
     {α, β, γ} = {0, 0, 0},
     {α, β, γ} = Re[{α, β, γ}];
   ];
   {α, β, γ}
  ];
```