In[1]:=
```
(* Set the folder for saving gifs *)
SetDirectory@NotebookDirectory[];

(*Some 2D vector math simple constructs helpers*)
```

In[2]:=
```
rotateVector[v_] := Join[{v[[2]], -v[[1]]}, Take[v, {3, -1}]]
normalize[v_] := Join[Take[v, {1, 2}] / Norm[Take[v, {1, 2}]], Take[v, {3, -1}]]
distanceToPlane[v_, plane_] := Join[v, {1}].plane
intersectTwoPlanes[p1_, p2_] :=
 {(p1[[2]] * p2[[3]] - p1[[3]] * p2[[2]]) / (p1[[1]] * p2[[2]] - p1[[2]] * p2[[1]]),
  (p1[[3]] * p2[[1]] - p1[[1]] * p2[[3]]) / (p1[[1]] * p2[[2]] - p1[[2]] * p2[[1]])}
projectPointOntoPlane[plane_, point_] := point - Join[point, {1}].plane * plane[[{1, 2}]]

(* Some cone / 2D triangle helper functions*)
```

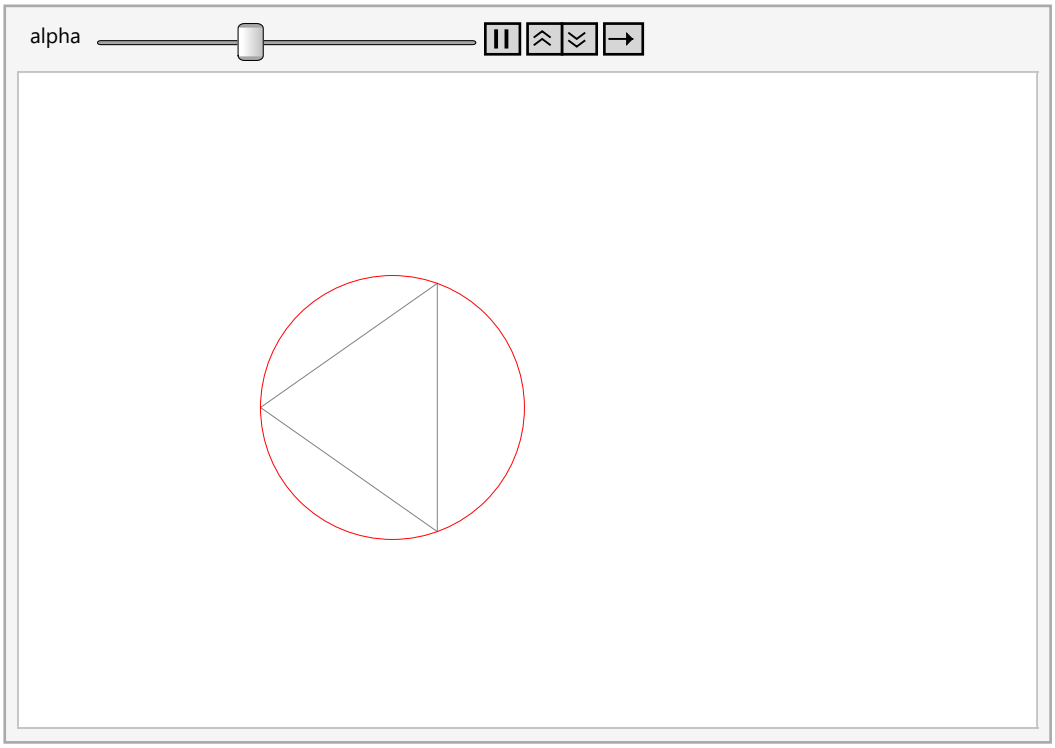
In[7]:=
```
coneCenter[l_] := l[[1]]
coneAngle[l_] := l[[2]]
coneRotation[l_] := l[[3]]
coneLength[l_] := l[[4]]
rotateVector[v_] := Join[{v[[2]], -v[[1]]}, Take[v, {3, -1}]]
makeCone[center_, angle_, rotation_, length_] := {center, angle, rotation, length}
coneCorner[cone_, first_: True] :=
 coneCenter[cone] + coneLength[cone] {Cos[coneAngle[cone]],
     Sin[coneAngle[cone]] * If[first, 1, -1] }.RotationMatrix[coneRotation[cone]]
coneForward[cone_] := {Cos[coneRotation[cone]], -Sin[coneRotation[cone]]}
```

In[15]:=
```
closestConePlanePoint[cone_, plane_] :=
 Module[{corner1, corner2, origin}, corner1 = coneCorner[cone];
  corner2 = coneCorner[cone, False];
  origin = coneCenter[cone];
  MinimalBy[{corner1, corner2, origin}, Function[distanceToPlane[#, plane]]][[1]]]

coneBoundingCircle[cone_] := If[coneAngle[cone] > π / 4,
  {coneCenter[cone] + coneForward[cone] * Cos[coneAngle[cone]], Sin[coneAngle[cone]]},
  {coneCenter[cone] + coneForward[cone] * 1 / (2 Cos[coneAngle[cone]]),
   1 / (2 Cos[coneAngle[cone]])}]
```

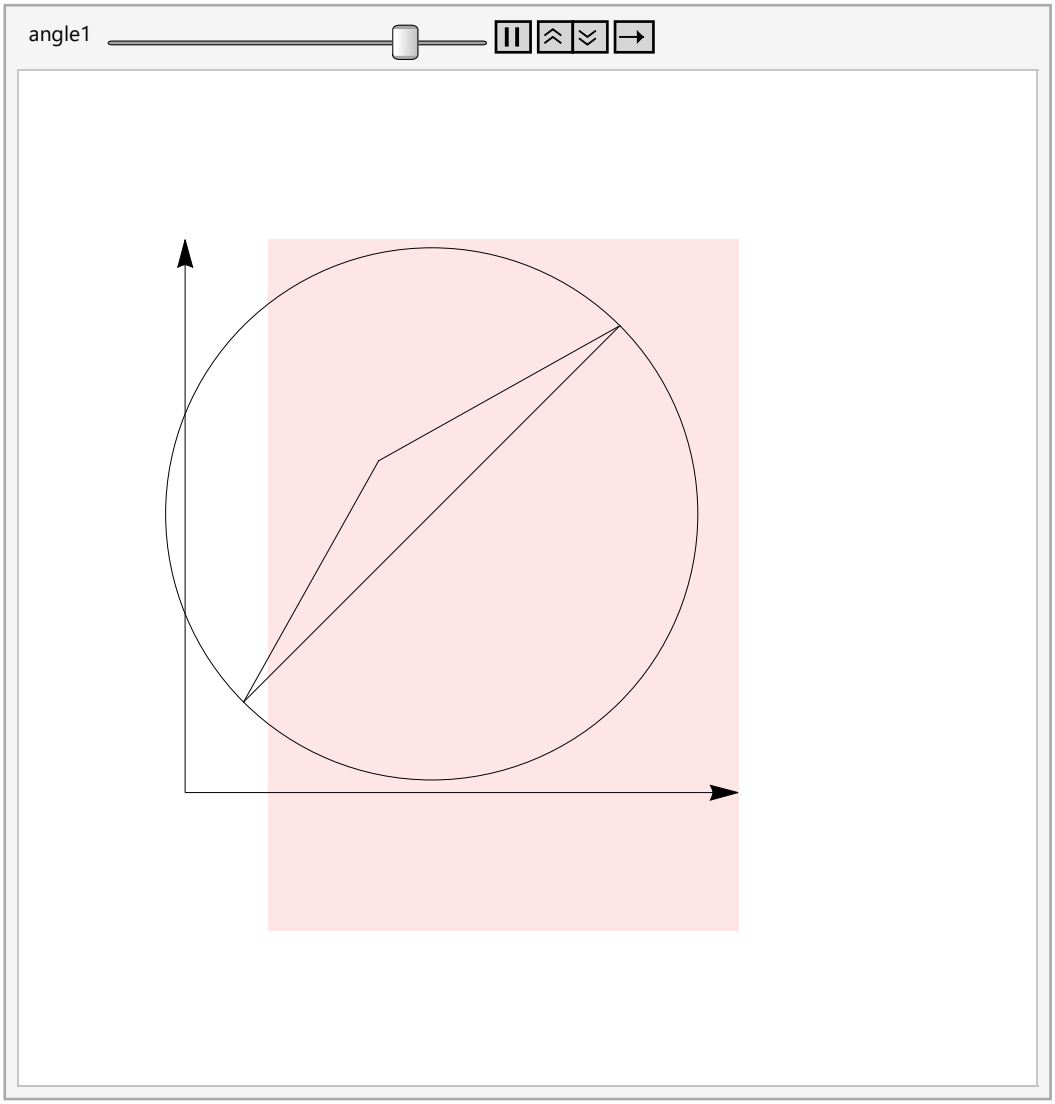
In[17]:=
```
mapDistances[cone_, plane_] :=
 Module[{corner1, corner2, origin}, corner1 = coneCorner[cone];
  corner2 = coneCorner[cone, False];
  origin = coneCenter[cone];
  Map[Function[distanceToPlane[#, plane]], {corner1, corner2, origin}]]

(*Collision detection*)
```

```
boundingBox[points_] := Module[{bb = CoordinateBoundingBox[points]},
  {{(bb[[1, 1]] + bb[[2, 1]]) * 0.5, (bb[[1, 2]] + bb[[2, 2]]) * 0.5},
   {(bb[[2, 1]] - bb[[1, 1]]) * 0.5, (bb[[2, 2]] - bb[[1, 2]]) * 0.5}}]
testSphereVsPlane[sphere_, plane_] :=
 If[distanceToPlane[sphere[[1]], plane] < sphere[[2]], True, False]
sphereVsBoundingBox[sphere_, bb_] :=
 Norm[Abs[sphere[[1]] - bb[[1]]] - bb[[2]]] < sphere[[2]]
testConeVsPlane[cone_, plane_] :=
 If[distanceToPlane[closestConePlanePoint[cone, plane], plane] < 0, True, False]

(*Some 2D visualizations*)

In[22]:= planeGraphics[v_] := Module[{vNorm, pointOnPlane}, vNorm = normalize[v][[{1, 2}]];
  pointOnPlane = vNorm * -v[[3]];
  {Opacity[0.1], HalfPlane[
    {pointOnPlane + rotateVector[vNorm], pointOnPlane - rotateVector[vNorm]}, vNorm]}]
coneGraphics[cone_] := {Line[{coneCenter[cone], coneCorner[cone]}],
  Line[{coneCenter[cone], coneCorner[cone, False]}],
  Line[{coneCorner[cone, False], coneCorner[cone, True]}]}
circleGraphics[sphere_] := {Circle[sphere[[1]], sphere[[2]]]}
boundingBoxGraphics[bb_] := Rectangle[bb[[1]] - bb[[2]], bb[[1]] + bb[[2]]]

(*First technique - cone bounding sphere *)
```

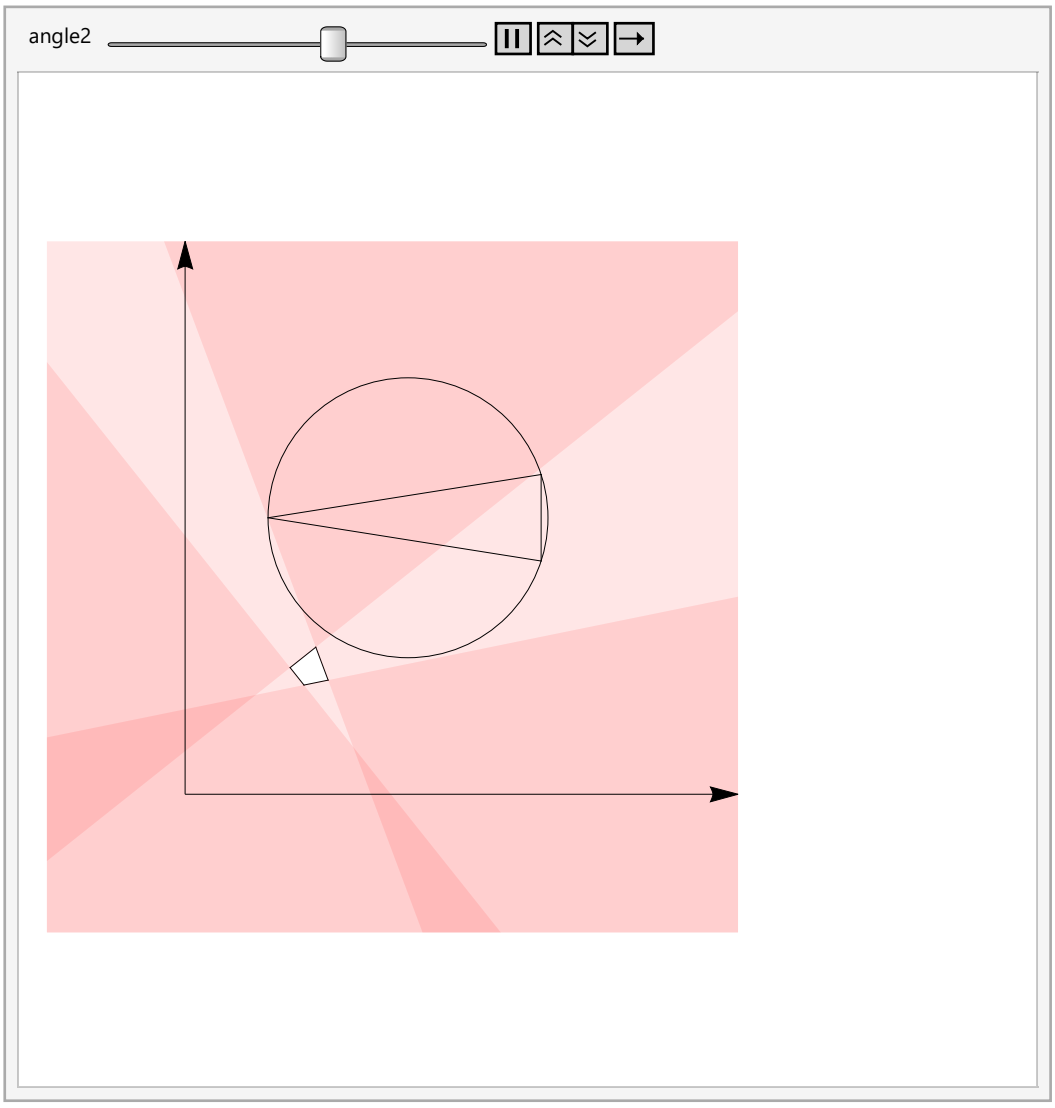
```
Animate[Module[{boundingSphere = coneBoundingCircle[makeCone[{0, 0}, alpha, 0, 1]]},
  Graphics[{Gray, coneGraphics[makeCone[{0, 0}, alpha, 0, 1]],
    Red, Circle[boundingSphere[[1]], boundingSphere[[2]]]},
   PlotRange → {{-1.6 + boundingSphere[[1, 1]], 1.6 + boundingSphere[[1, 1]]},
     {-1, 1}}]], {alpha, 0, π / 2}]
```



```
(*Cone bounding sphere vs plane *)
Animate[Module[{testCone, testPlane, testCircle, testColor},
  testCone = makeCone[{0.7, 1.2}, angle1, π / 4, 1];
  testPlane = normalize[{Cos[0], Sin[0], -{Cos[0], Sin[0]}.{0.3, 0.5}}];
  testCircle = coneBoundingCircle[testCone];
  testColor = If[testSphereVsPlane[testCircle, testPlane], Red, Green];
  Show[Graphics[{testColor, planeGraphics[testPlane]}],
   Graphics[circleGraphics[testCircle]],
   Graphics[coneGraphics[testCone]],
   Graphics[{Black, Arrow[{{0, 0}, {0, 2}}], Arrow[{{0, 0}, {2, 0}}]}],
   PlotRange → {{-0.5, 2}, {-0.5, 2}}]], {angle1, 0, π/2}]
```

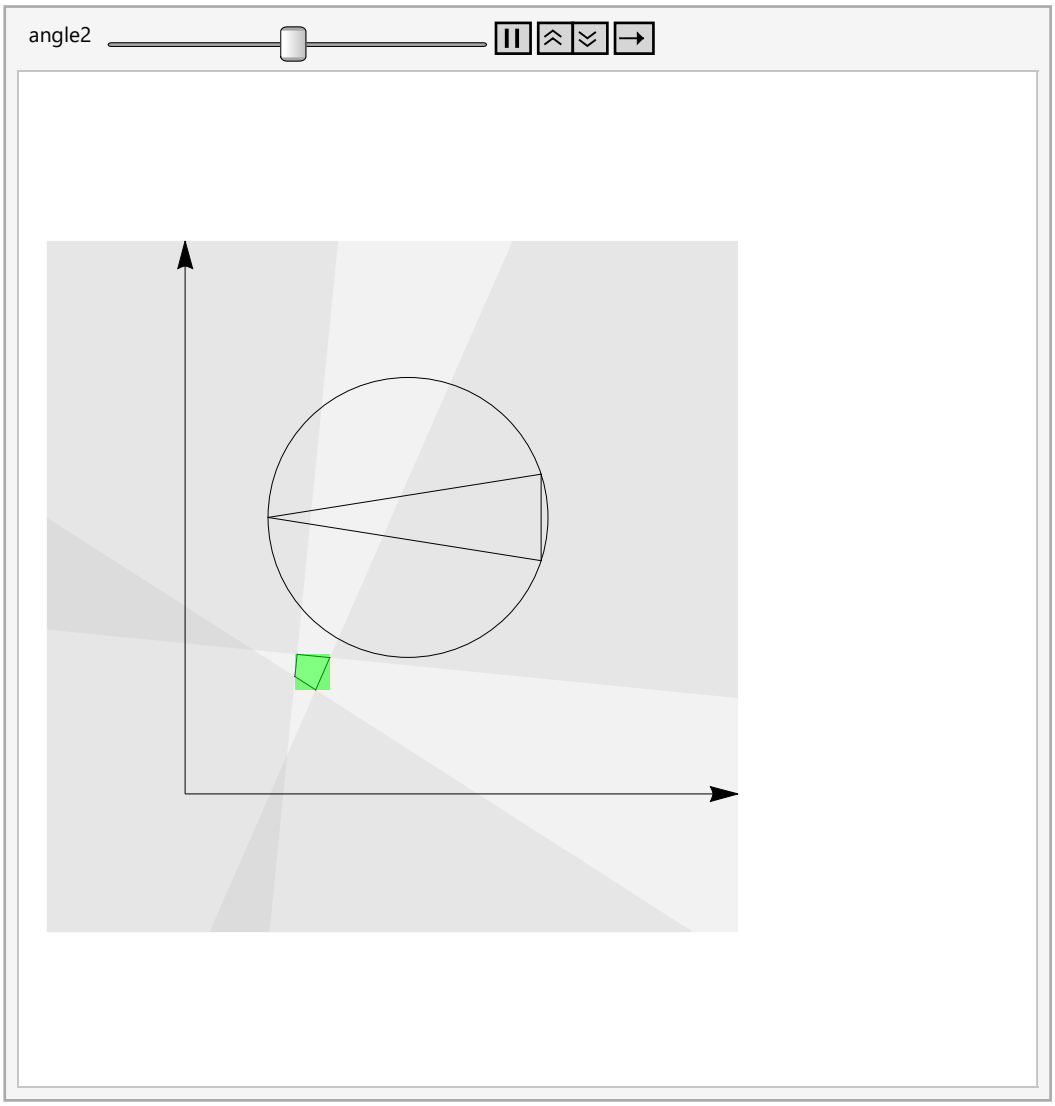


```
(*Cone bounding sphere vs frustum and few planes*)
Animate[Module[{testCone, angles, testCircle, planes},
  testCone = makeCone[-{Cos[0], Sin[0]} * 0.5 + {0.8, 1.0}, π/20, 0, 1.0];
  angles = {angle2, angle2 + π/2 * 0.7, angle2 + π * 0.9, angle2 + π * 3/2};
  testCircle = coneBoundingCircle[testCone];
  planes = Map[Function[
     normalize[{Cos[#], Sin[#], -{Cos[#], Sin[#]}.{0.45, 0.45} - 0.05}]], angles];
  Show[Map[Function[Graphics[{Line[{intersectTwoPlanes[#[[1]], #[[2]]],
        intersectTwoPlanes[#[[2]], #[[3]]]}]}]], Partition[planes, 3, 1, 1]],
   Map[Function[Graphics[{If[testSphereVsPlane[testCircle, #], Red, Green],
      planeGraphics[#]}]], planes],
   Graphics[coneGraphics[testCone]], Graphics[circleGraphics[testCircle]],
   Graphics[{Black, Arrow[{{0, 0}, {0, 2}}], Arrow[{{0, 0}, {2, 0}}]}],
   PlotRange → {{-0.5, 2}, {-0.5, 2}}]], {angle2, 0, 2 π}]
```



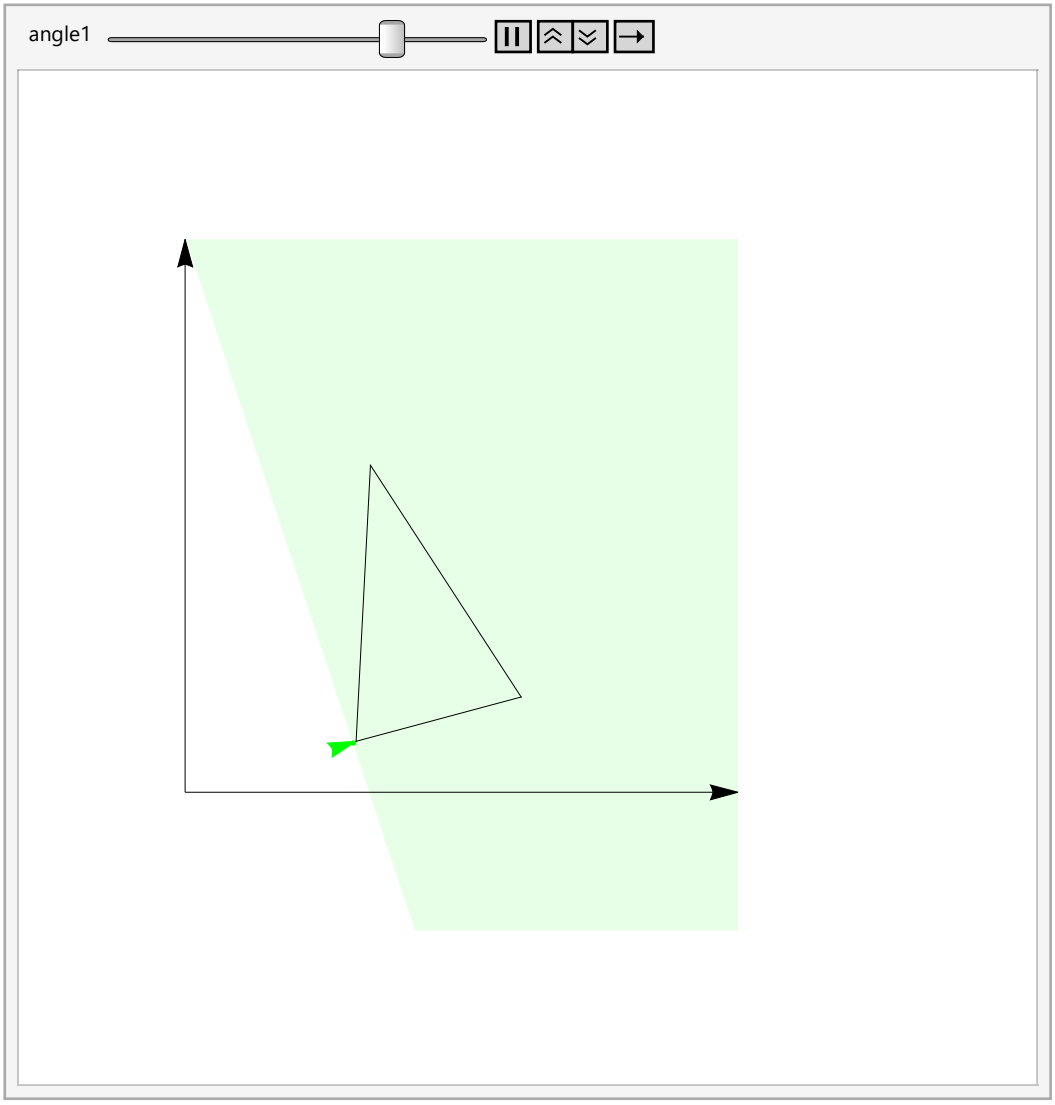

```
(*Frustum bounding box vs bounding sphere *)
```

```
Animate[Module[{testCone, angles, testCircle, planes, points},
  testCone = makeCone[-{Cos[0], Sin[0]} * 0.5 + {0.8, 1.0}, π/20, 0, 1.0];
  angles = {angle2, angle2 + π/2 * 0.7, angle2 + π * 0.9, angle2 + π * 3/2};
  testCircle = coneBoundingCircle[testCone];
  planes = Map[
    Function[normalize[{Cos[#], Sin[#], -{Cos[#], Sin[#]}.{0.45, 0.45} - 0.05}]], angles];
  points = Map[Function[intersectTwoPlanes[#[[1]], #[[2]]]], Partition[planes, 2, 1, 1]];
  Show[Map[Function[Graphics[{Line[{intersectTwoPlanes[#[[1]], #[[2]]],
        intersectTwoPlanes[#[[2]], #[[3]]]}]}]], Partition[planes, 3, 1, 1]],
   Map[Function[Graphics[{Gray, planeGraphics[#]}]], planes],
   Graphics[{Green, Opacity[0.5], boundingBoxGraphics[boundingBox[points]]}],
   Graphics[coneGraphics[testCone]], Graphics[circleGraphics[testCircle]],
   Graphics[{Black, Arrow[{{0, 0}, {0, 2}}], Arrow[{{0, 0}, {2, 0}}]}],
   PlotRange → {{-0.5, 2}, {-0.5, 2}}]], {angle2, 0, 2 π}]
```

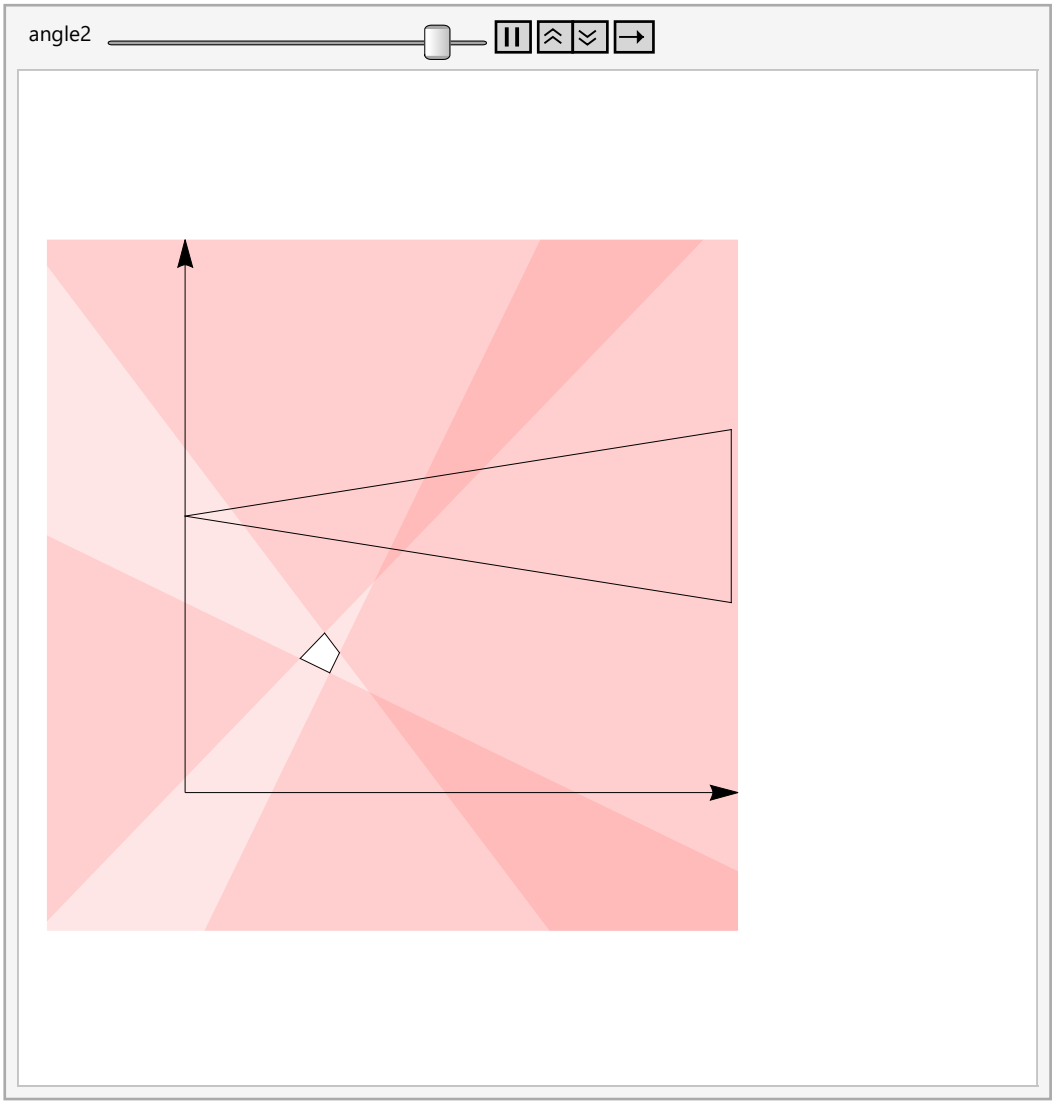



```
(*Show cone vs plane precise test *)
```

```
Animate[Module[{testCone, testPlane, closestTestPoint, testColor},
  testCone = makeCone[-{Cos[angle1], Sin[angle1]} * 0.5 + {0.8, 0.7}, π/10, -angle1, 1];
  testPlane = normalize[{Cos[0.32], Sin[0.32], -{Cos[0.32], Sin[0.32]}.{0.5, 0.5}}];
  closestTestPoint = closestConePlanePoint[testCone, testPlane];
  testColor = If[testConeVsPlane[testCone, testPlane], Red, Green];
  Show[Graphics[{testColor, planeGraphics[testPlane]}],
   Graphics[{testColor,
     Arrow[{projectPointOntoPlane[testPlane, closestTestPoint], closestTestPoint}]}],
   Graphics[coneGraphics[testCone]], Graphics[{Black, Arrow[{{0, 0}, {0, 2}}],
     Arrow[{{0, 0}, {2, 0}}]}], PlotRange → {{-0.5, 2}, {-0.5, 2}}]], {angle1, 0, 2 π}]
```

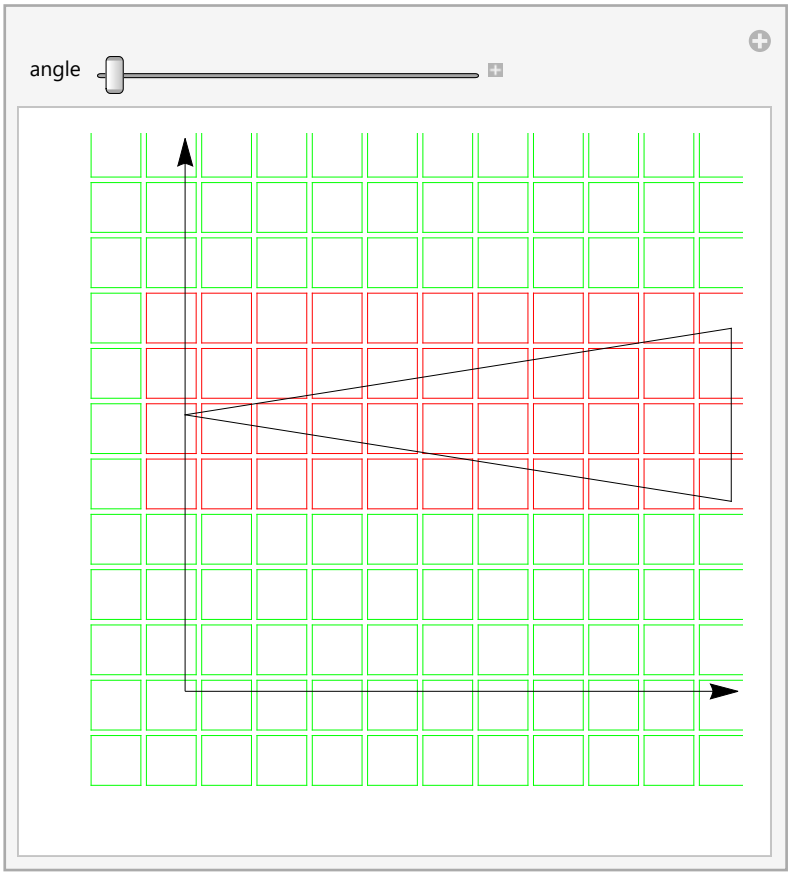


```
(* Cone vs frustum test *)
Animate[Module[{testCone, angles, planes},
  testCone = makeCone[-{Cos[0], Sin[0]} * 0.5 + {0.5, 1.0}, π/20, -0, 2.0];
  angles = {angle2, angle2 + π/2 * 0.7, angle2 + π * 0.9, angle2 + π * 3/2};
  planes = Map[Function[
     normalize[{Cos[#], Sin[#], -{Cos[#], Sin[#]}.{0.5, 0.5} + -0.05}]], angles];
  Show[Map[Function[Graphics[{Line[{intersectTwoPlanes[#[[1]], #[[2]]],
        intersectTwoPlanes[#[[2]], #[[3]]]}]}]], Partition[planes, 3, 1, 1]], Map[
    Function[Graphics[{If[testConeVsPlane[testCone, #], Red, Green], planeGraphics[#]}]],
    planes],
   Graphics[coneGraphics[testCone]], Graphics[{Black, Arrow[{{0, 0}, {0, 2}}],
     Arrow[{{0, 0}, {2, 0}}]}], PlotRange → {{-0.5, 2}, {-0.5, 2}}]], {angle2, 0, 2 π}]
```



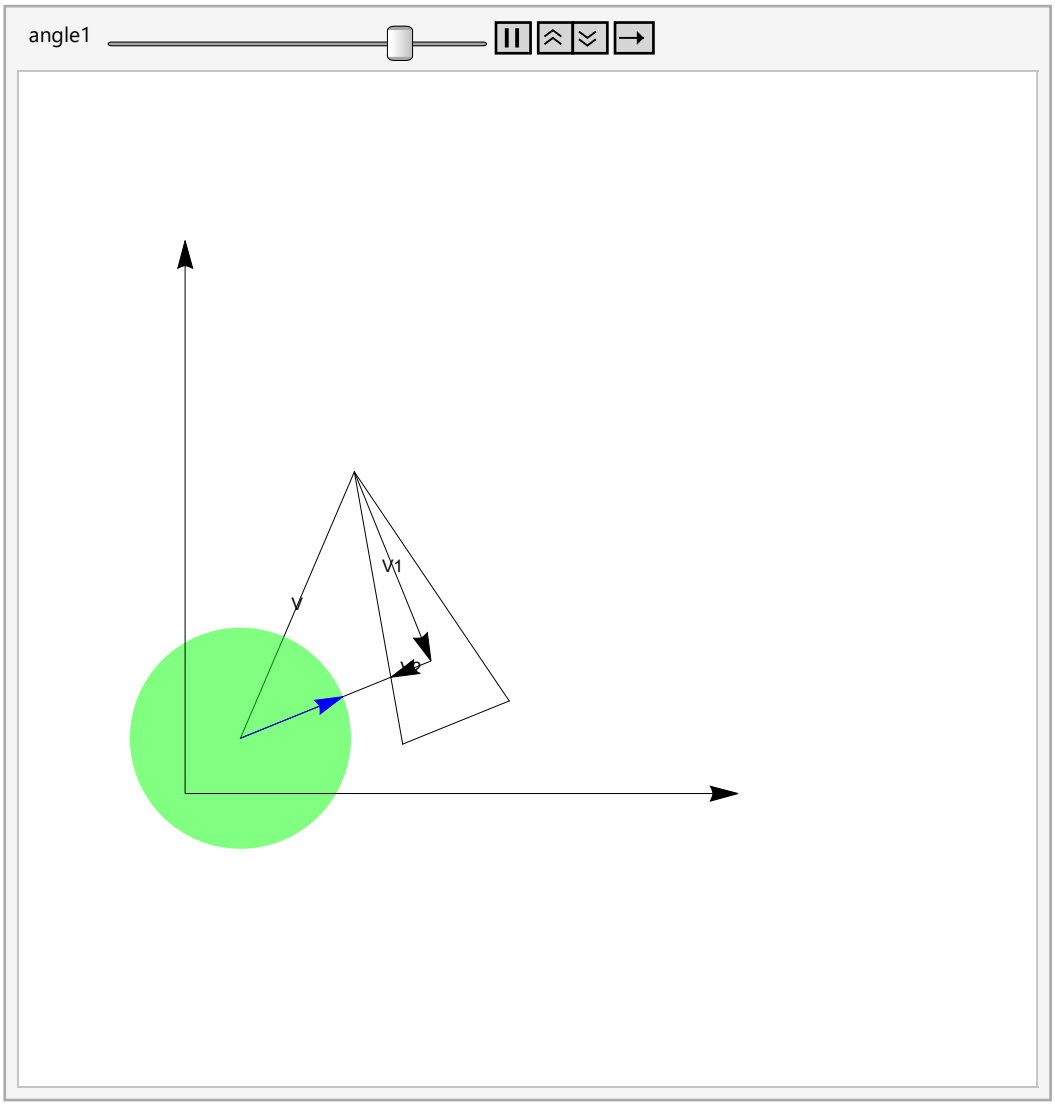

```
(*How bad is cone vs 4 planes test in practice? Showing with a grid*)
```

```
Manipulate[Module[{testCone, angles},
  testCone = makeCone[-{Cos[angle], Sin[angle]} * 0.5 + {0.5, 1.0}, π / 20, -angle, 2.0];
  angles = {0, 0 + π / 2, 0 + π, 0 + π * 3 / 2};
  Show[Table[Module[{planes = Map[Function[
        normalize[{Cos[#], Sin[#], -{Cos[#], Sin[#]}.{x, y} - 0.09}]], angles]},
    Map[Function[Graphics[{If[Total[Map[Function[If[testConeVsPlane[testCone,
              #], 1, 0]], planes]] == 4 , Red, Green], {Line[{intersectTwoPlanes[
          #[[1]], #[[2]]], intersectTwoPlanes[#[[2]], #[[3]]]}]}]]],
     Partition[planes, 3, 1, 1]]], {x, -0.25, 2, 0.2}, {y, -0.25, 2, 0.2}],
   Graphics[{coneGraphics[testCone]}], Graphics[{Black, Arrow[{{0, 0}, {0, 2}}],
    Arrow[{{0, 0}, {2, 0}}]}], PlotRange → {{-0.5, 2}, {-0.5, 2}}]], {angle, 0, 2 π}]
```



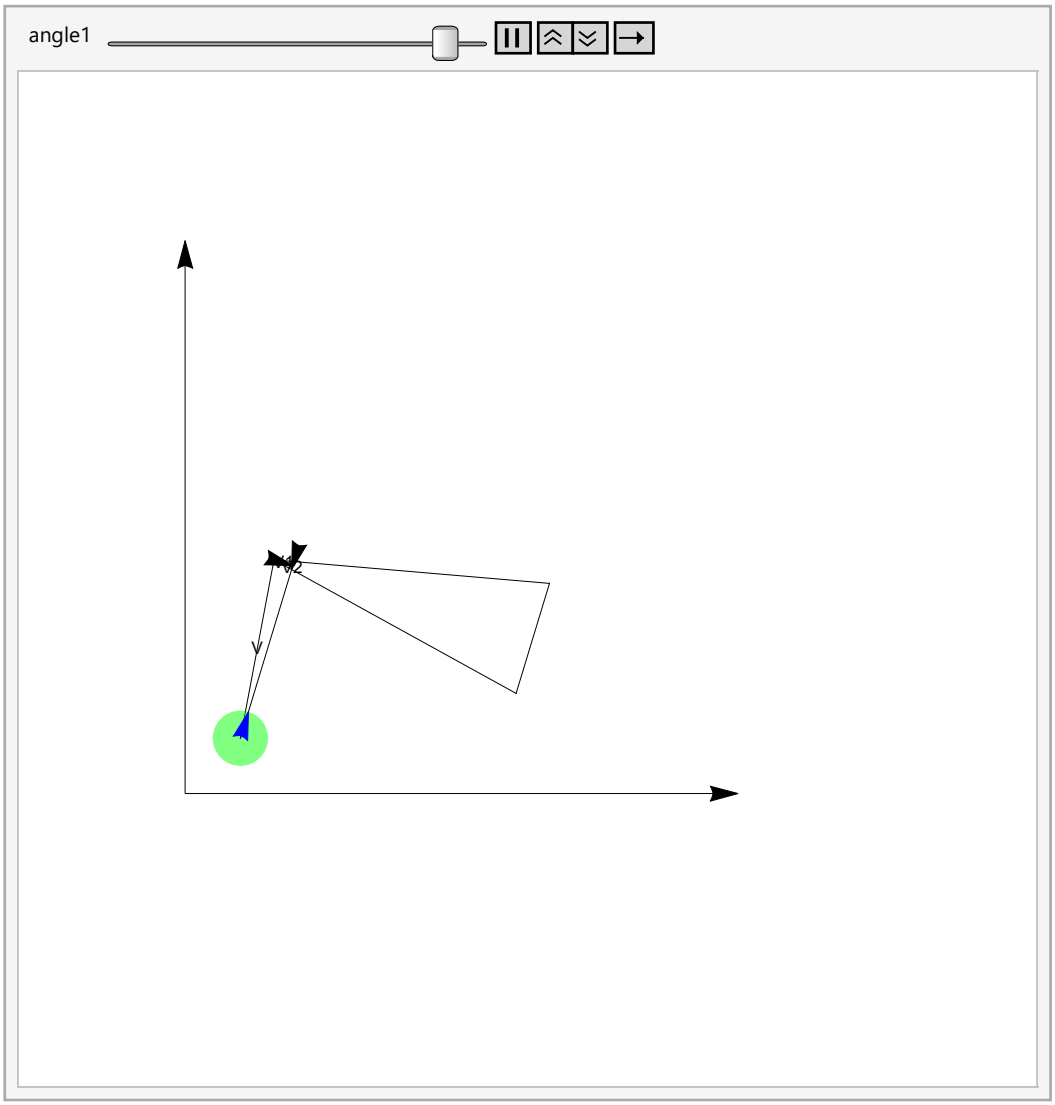

```
(*Show cone vs sphere test*)
```

```
Animate[
 Module[{testCone, testSphere, coneSpereAxis, coneSphereAxisPerp, closestConeSpherePoint},
  testCone = makeCone[-{Cos[angle1], Sin[angle1]} * 0.5 + {0.8, 0.7}, π/15, -angle1, 1];
  testSphere = {{0.2, 0.2}, 0.4};
  coneSpereAxis = coneForward[testCone] *
    Clip[(testSphere[[1]] - coneCenter[testCone]).coneForward[testCone], {0, 1}];
  coneSphereAxisPerp = testSphere[[1]] - coneCenter[testCone] - coneForward[testCone] *
     (testSphere[[1]] - coneCenter[testCone]).coneForward[testCone];
  coneSphereAxisPerp = Min[Norm[coneSphereAxisPerp],
     Norm[coneSpereAxis] * Tan[coneAngle[testCone]]] * Normalize[coneSphereAxisPerp];
  closestConeSpherePoint = coneSpereAxis + coneSphereAxisPerp + coneCenter[testCone];
  Show[Graphics[{Line[{testSphere[[1]], coneCenter[testCone]}],
     Text["V", (testSphere[[1]] + coneCenter[testCone]) * 0.5]}],
   Graphics[{If[Norm[closestConeSpherePoint - testSphere[[1]]] < testSphere[[2]],
      Red, Green], Opacity[0.5], Disk[testSphere[[1]], testSphere[[2]]]}],
   Graphics[{Arrow[{coneCenter[testCone], coneCenter[testCone] + coneSpereAxis}],
     Text["V1", coneCenter[testCone] + coneSpereAxis * 0.5]}],
   Graphics[{Arrow[{coneCenter[testCone] + coneSpereAxis, closestConeSpherePoint}],
     Text["V2", (coneCenter[testCone] + coneSpereAxis + closestConeSpherePoint) * 0.5]}],
   Graphics[Line[{testSphere[[1]], closestConeSpherePoint}]],
   Graphics[{Blue, Arrow[{testSphere[[1]],
       testSphere[[1]] + Normalize[closestConeSpherePoint - testSphere[[1]]] *
         Min[Norm[closestConeSpherePoint - testSphere[[1]]], testSphere[[2]]]}]}],
   Graphics[coneGraphics[testCone]], Graphics[{Black, Arrow[{{0, 0}, {0, 2}}],
     Arrow[{{0, 0}, {2, 0}}]}], PlotRange → {{-0.5, 2}, {-0.5, 2}}]], {angle1, 0, 2 π}]
```



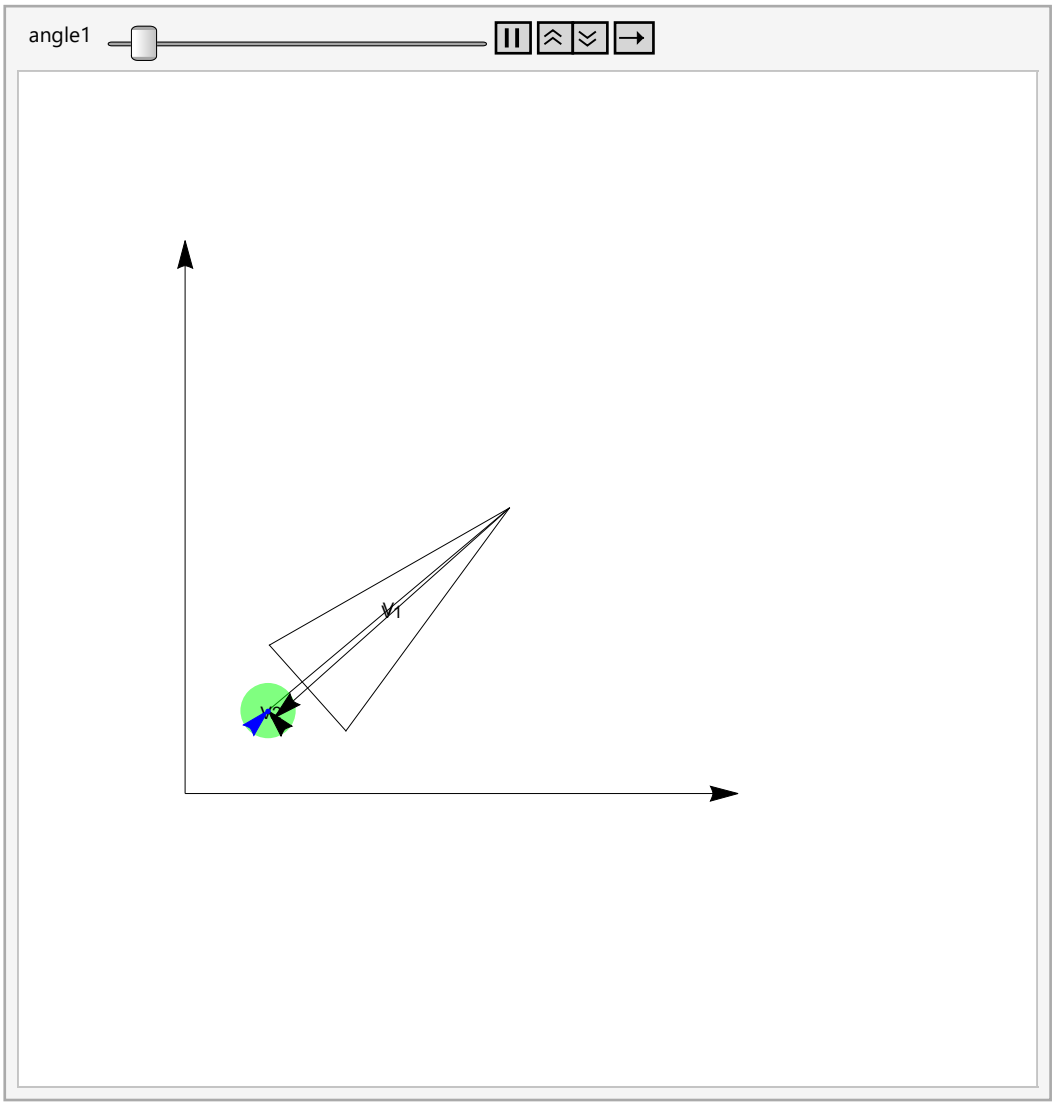

```
(*Charles Bloom simplified test*)
```

```
Animate[
 Module[{testCone, testSphere, coneSpereAxis, coneSphereAxisPerp, closestConeSpherePoint},
  testCone = makeCone[-{Cos[angle1], Sin[angle1]} * 0.5 + {0.8, 0.7}, π/15, -angle1, 1];
  testSphere = {{0.2, 0.2}, 0.1};
  coneSpereAxis =
   coneForward[testCone] * (testSphere[[1]] - coneCenter[testCone]).coneForward[testCone];
  coneSphereAxisPerp = testSphere[[1]] - coneCenter[testCone] - coneForward[testCone] *
     (testSphere[[1]] - coneCenter[testCone]).coneForward[testCone];
  coneSphereAxisPerp = Min[Norm[coneSphereAxisPerp],
     Norm[coneSpereAxis] * Tan[coneAngle[testCone]]] * Normalize[coneSphereAxisPerp];
  closestConeSpherePoint = coneSpereAxis + coneSphereAxisPerp + coneCenter[testCone];
  Show[Graphics[{Line[{testSphere[[1]], coneCenter[testCone]}],
     Text["V", (testSphere[[1]] + coneCenter[testCone]) * 0.5]}],
   Graphics[{If[Norm[closestConeSpherePoint - testSphere[[1]]] < testSphere[[2]],
      Red, Green], Opacity[0.5], Disk[testSphere[[1]], testSphere[[2]]]}],
   Graphics[{Arrow[{coneCenter[testCone], coneCenter[testCone] + coneSpereAxis}],
     Text["V1", coneCenter[testCone] + coneSpereAxis * 0.5]}],
   Graphics[{Arrow[{coneCenter[testCone] + coneSpereAxis, closestConeSpherePoint}],
     Text["V2", (coneCenter[testCone] + coneSpereAxis + closestConeSpherePoint) * 0.5]}],
   Graphics[Line[{testSphere[[1]], closestConeSpherePoint}]],
   Graphics[{Blue, Arrow[{testSphere[[1]],
       testSphere[[1]] + Normalize[closestConeSpherePoint - testSphere[[1]]] *
         Min[Norm[closestConeSpherePoint - testSphere[[1]]], testSphere[[2]]]}]}],
   Graphics[coneGraphics[testCone]], Graphics[{Black, Arrow[{{0, 0}, {0, 2}}],
     Arrow[{{0, 0}, {2, 0}}]}], PlotRange → {{-0.5, 2}, {-0.5, 2}}]], {angle1, 0, 2 π}]
```



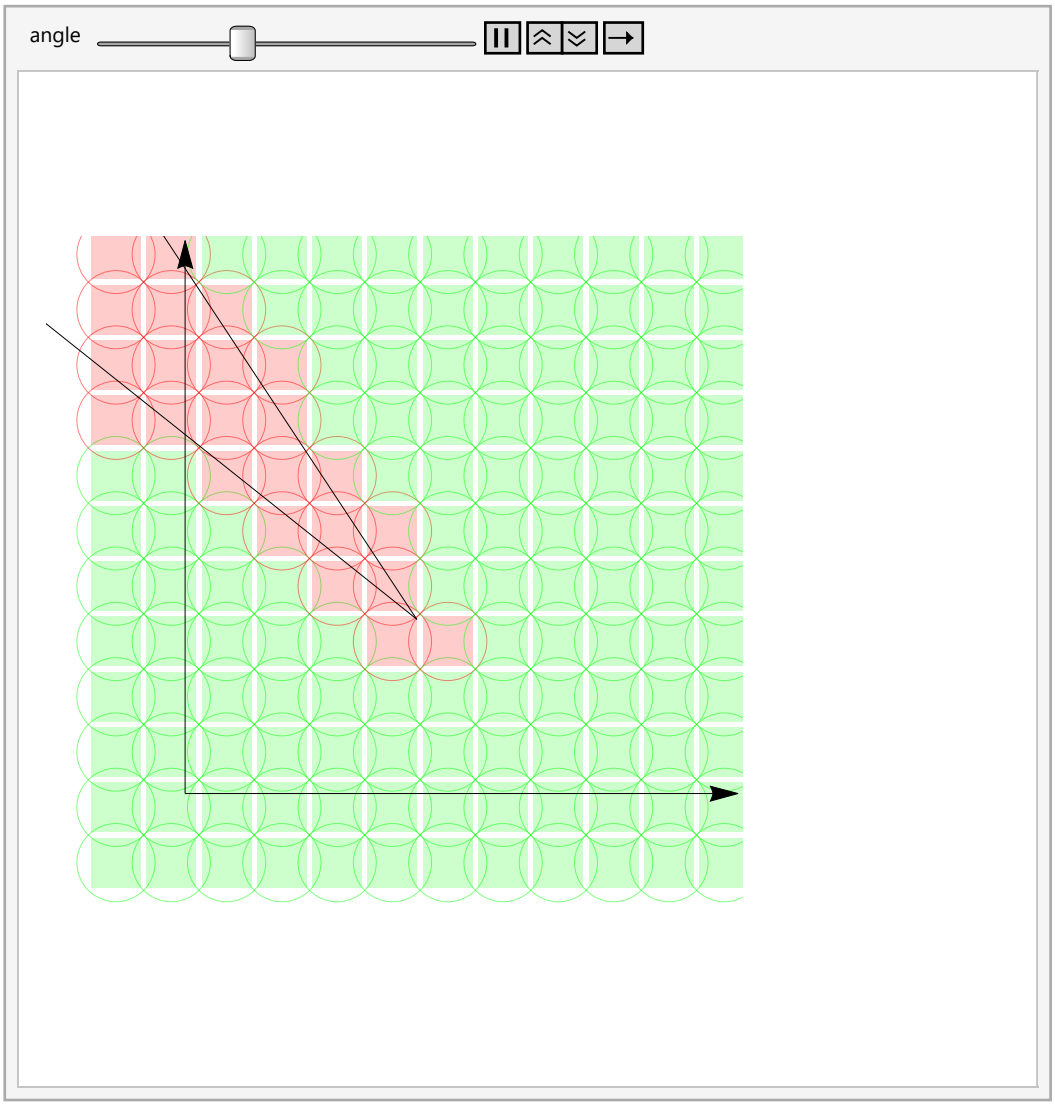

```
(*Corrected, still very cheap version*)
```

```
Animate[Module[{testCone, testSphere, coneSpereAxisL,
    coneSpereAxis, coneSphereAxisPerp, closestConeSpherePoint},
  testCone = makeCone[-{Cos[angle1], Sin[angle1]} * 0.5 + {0.8, 0.7}, π/15, -angle1, 1];
  testSphere = {{0.3, 0.3}, 0.1};
  coneSphereAxisL = (testSphere[[1]] - coneCenter[testCone]).coneForward[testCone];
  coneSpereAxis =
   coneForward[testCone] * (testSphere[[1]] - coneCenter[testCone]).coneForward[testCone];
  coneSphereAxisPerp = testSphere[[1]] - coneCenter[testCone] - coneForward[testCone] *
     (testSphere[[1]] - coneCenter[testCone]).coneForward[testCone];
  coneSphereAxisPerp = Min[Norm[coneSphereAxisPerp],
     Norm[coneSpereAxis] * Tan[coneAngle[testCone]]] * Normalize[coneSphereAxisPerp];
  closestConeSpherePoint = coneSpereAxis + coneSphereAxisPerp + coneCenter[testCone];
  Show[Graphics[{Line[{testSphere[[1]], coneCenter[testCone]}],
     Text["V", (testSphere[[1]] + coneCenter[testCone]) * 0.5]}],
   Graphics[{If[Norm[closestConeSpherePoint - testSphere[[1]]] < testSphere[[2]] &&
       coneSphereAxisL > - testSphere[[2]] &&
       coneSphereAxisL < coneLength[testCone] + testSphere[[2]], Red, Green],
     Opacity[0.5], Disk[testSphere[[1]], testSphere[[2]]]}],
   Graphics[{Arrow[{coneCenter[testCone], coneCenter[testCone] + coneSpereAxis}],
     Text["V1", coneCenter[testCone] + coneSpereAxis * 0.5]}],
   Graphics[{Arrow[{coneCenter[testCone] + coneSpereAxis, closestConeSpherePoint}],
     Text["V2", (coneCenter[testCone] + coneSpereAxis + closestConeSpherePoint) * 0.5]}],
   Graphics[Line[{testSphere[[1]], closestConeSpherePoint}]],
   Graphics[{Blue, Arrow[{testSphere[[1]],
       testSphere[[1]] + Normalize[closestConeSpherePoint - testSphere[[1]]] *
         Min[Norm[closestConeSpherePoint - testSphere[[1]]], testSphere[[2]]]}]}],
   Graphics[coneGraphics[testCone]], Graphics[{Black, Arrow[{{0, 0}, {0, 2}}],
     Arrow[{{0, 0}, {2, 0}}]}], PlotRange → {{-0.5, 2}, {-0.5, 2}}], {angle1, 0, 2 π}]
```

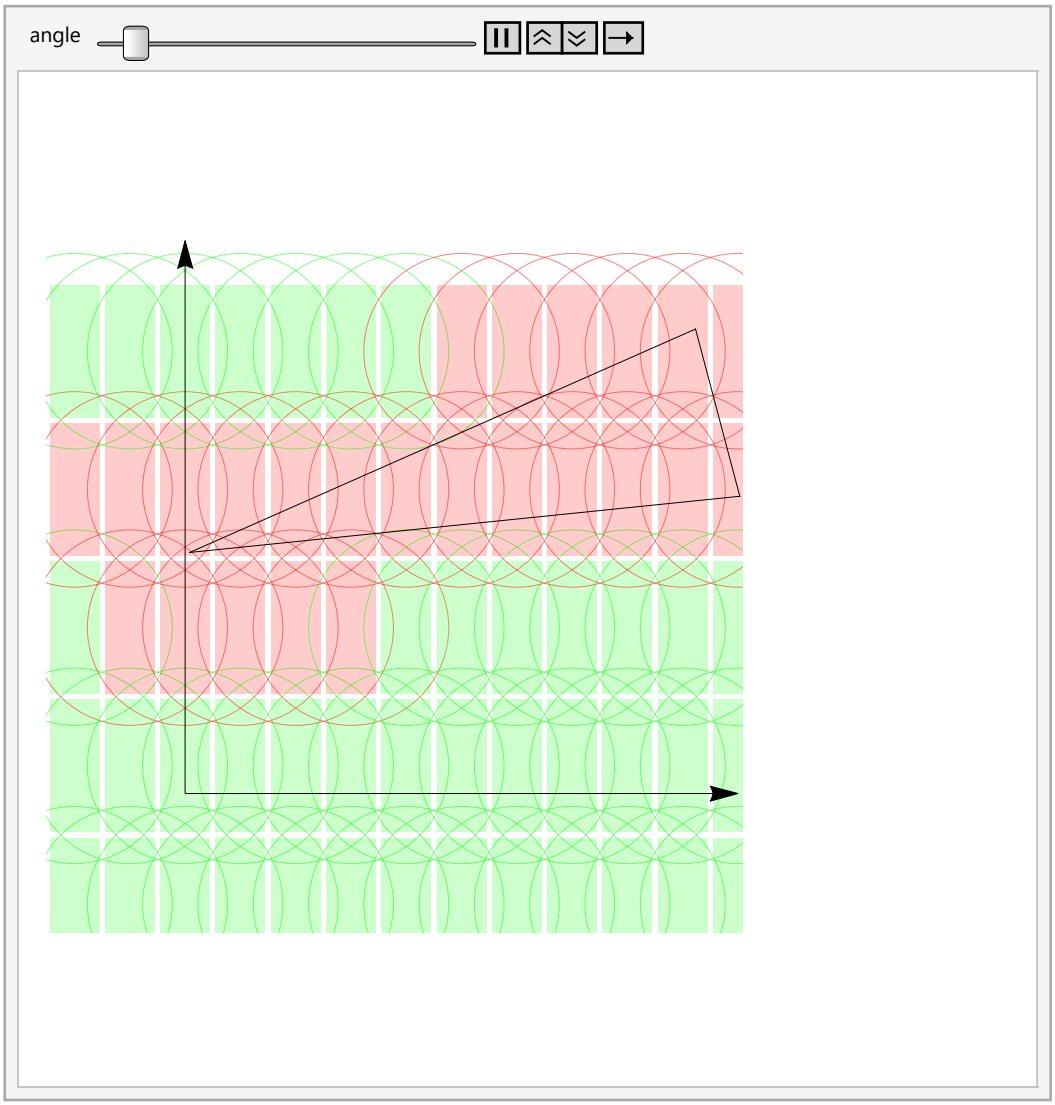



```
(*Cone vs sphere on grid of vertices – cheap / approx test *)
```

```
Animate[Module[{testCone, angles},
  testCone = makeCone[-{Cos[angle], Sin[angle]} * 0.5 + {0.5, 1.0}, π / 20, -angle, 2.0];
  Show[Table[Module[{testSphere, coneSphereAxisL, coneSpereAxis, coneSphereAxisPerp,
      closestConeSpherePoint}, testSphere = {{x, y}, 0.1 * Sqrt[2]};
     coneSphereAxisL = (testSphere[[1]] - coneCenter[testCone]).coneForward[testCone];
     coneSpereAxis = coneForward[testCone] *
       (testSphere[[1]] - coneCenter[testCone]).coneForward[testCone];
     coneSphereAxisPerp = testSphere[[1]] - coneCenter[testCone] - coneForward[testCone] *
       (testSphere[[1]] - coneCenter[testCone]).coneForward[testCone];
     coneSphereAxisPerp = Min[Norm[coneSphereAxisPerp],
        Norm[coneSpereAxis] * Tan[coneAngle[testCone]]] * Normalize[coneSphereAxisPerp];
     closestConeSpherePoint = coneSpereAxis + coneSphereAxisPerp + coneCenter[testCone];
     Graphics[{If[Norm[closestConeSpherePoint - testSphere[[1]]] < testSphere[[2]] &&
         coneSphereAxisL > - testSphere[[2]] &&
         coneSphereAxisL < coneLength[testCone] + testSphere[[2]], Red, Green],
       Opacity[0.5], Circle[testSphere[[1]], testSphere[[2]]], Opacity[0.2],
       Rectangle[{x, y} - 0.09, {x, y} + 0.09]}]], {x, -0.25, 2, 0.2}, {y, -0.25, 2, 0.2}],
   Graphics[{coneGraphics[testCone]}], Graphics[{Black, Arrow[{{0, 0}, {0, 2}}],
     Arrow[{{0, 0}, {2, 0}}]}], PlotRange → {{-0.5, 2}, {-0.5, 2}}]], {angle, 0, 2 π}]
```


angle

```
(*Same test, long frusta*)
Animate[Module[{testCone, angles},
  testCone = makeCone[-{Cos[angle], Sin[angle]} * 0.5 + {0.5, 1.0}, π/20, -angle, 2.0];
  Show[Table[Module[{testSphere, coneSphereAxisL, coneSpereAxis, coneSphereAxisPerp,
      closestConeSpherePoint}, testSphere = {{x, y}, 0.25 * Sqrt[2]};
    coneSphereAxisL = (testSphere[[1]] - coneCenter[testCone]).coneForward[testCone];
    coneSpereAxis = coneForward[testCone] *
      (testSphere[[1]] - coneCenter[testCone]).coneForward[testCone];
    coneSphereAxisPerp = testSphere[[1]] - coneCenter[testCone] - coneForward[testCone] *
      (testSphere[[1]] - coneCenter[testCone]).coneForward[testCone];
    coneSphereAxisPerp = Min[Norm[coneSphereAxisPerp],
      Norm[coneSpereAxis] * Tan[coneAngle[testCone]]] * Normalize[coneSphereAxisPerp];
    closestConeSpherePoint = coneSpereAxis + coneSphereAxisPerp + coneCenter[testCone];
    Graphics[{If[Norm[closestConeSpherePoint - testSphere[[1]]] < testSphere[[2]] &&
       coneSphereAxisL > - testSphere[[2]] &&
       coneSphereAxisL < coneLength[testCone] + testSphere[[2]], Red, Green],
     Opacity[0.5], Circle[testSphere[[1]], testSphere[[2]]], Opacity[0.2],
     Rectangle[{x, y} - {0.09, 0.24}, {x, y} + {0.09, 0.24}]}]],
   {x, -0.4, 2, 0.2}, {y, -0.4, 2, 0.5}],
  Graphics[{coneGraphics[testCone]}], Graphics[{Black, Arrow[{{0, 0}, {0, 2}}],
    Arrow[{{0, 0}, {2, 0}}]}], PlotRange → {{-0.5, 2}, {-0.5, 2}}]], {angle, 0, 2 π}]
```


angle