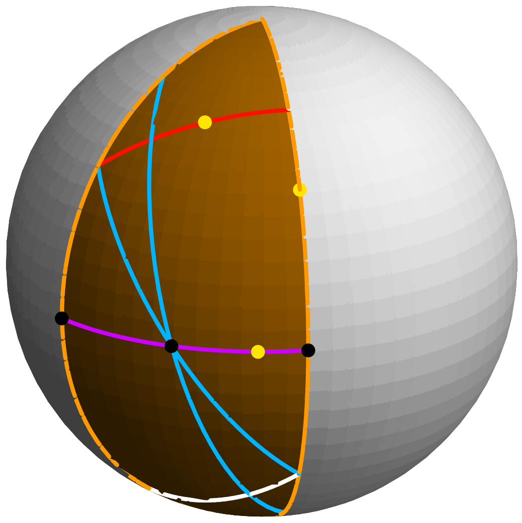
```
In[89]:= Paxes[t1_, t2_, t3_, u_, c_, lengthAxis_,
    BBrad_, ArrowScale_, tkns_, eye_, WantArrowColor_] := {
    If[t1, {{Thickness[tkns],    Line[c + u.# & /@ {{BBrad, 0, 0}, {.95 lengthAxis, 0, 0}}]},
      {Lighting → "Neutral", If[WantArrowColor, Hue[.01], {}],
       Arrow3D[c, c + 1.07 u.{lengthAxis, 0, 0}, ArrowScale, .1 tkns, eye]}}, {}],
    If[t2, {{Thickness[tkns],    Line[c + u.# & /@ {{0, BBrad, 0}, {0, .95 lengthAxis, 0}}]},
      {Lighting → "Neutral", If[WantArrowColor, Hue[.6], {}],
       Arrow3D[c, c + 1.07 u.{0, lengthAxis, 0}, ArrowScale, .1 tkns, eye]}}, {}],
    If[t3, {{Thickness[tkns],    Line[c + u.# & /@ {{0, 0, BBrad}, {0, 0, .95 lengthAxis}}]},
      {Lighting → "Neutral", If[WantArrowColor, Hue[.15], {}],
       Arrow3D[c, c + 1.07 u.{0, 0, lengthAxis}, ArrowScale, .1 tkns, eye]}}, {}]};

In[90]:= Lune[Λlist_, ExtraPts_, eye_, Λaxes_, uvwAxes_, grid_,
   ZeroLam1Lam3_, DeviatoricArc_, LuneBoundary_, νList_] := Graphics3D[{
    If[True, {GrayLevel[.7], EdgeForm[], Polygon /@ SpherePolyList[1, 30, 330, 0, 180, 5, 5]},
     {}],     (* opaque sphere *)
    If[True, {Hue[.1, 1, .5], EdgeForm[], Polygon /@ Map[MapEyeward[#, -.015, eye] &,
       SpherePolyList[1, -30, 30, 0, 180, 5, 5], {2}]}, {}],   (* opaque lune *)
    If[uvwAxes, Paxes[True, True, True, id, {0, 0, 0}, 1.3, 1, ArrowScale, tkns, eye, False],
     {}],    (* uvw axes *)
    If[Λaxes,   Paxes[True, True, True, uG, {0, 0, 0}, 1.3, 1, ArrowScale, tkns, eye, True],
     {}],      (* Λ axes *)
    If[grid, {Hue[.10], Table[Line[Table[xyztp[{γ, β}], {γ, -30, 30, 2.}]],
       {β, 10, 170, 10.}],     (* lunar lat-long grid *)
                         Table[Line[Table[xyztp[{γ, β}], {β, 10, 170, 2.}]],
       {γ, -30, 30, 10.}]}, {}],

    If[ZeroLam1Lam3, {{Hue[.01], Thickness[tkns],
       Line[arc[uG.{1, 0, 0}, uG.unit[{1, 1, 0}]]]},    (* zero contour for λ1 and λ3 *)
                        {White,  Thickness[tkns],
       Line[arc[uG.{0, 0, -1}, uG.unit[{0, -1, -1}]]]} }, {}],
    If[DeviatoricArc, {Hue[.80], Thickness[tkns],
      Line[arc[uG.unit[{2, -1, -1}], uG.unit[{1, 1, -2}]]]}, {}],    (* deviatoric arc *)
    If[LuneBoundary,  {Hue[.10], Thickness[tkns],
      Line[xyztp[{-30, #}] & /@ Range[0., 180, 3]],

      Line[xyztp[{ 30, #}] & /@ Range[0., 180, 3]]}, {}],    (* lune boundary curves *)

    {Hue[.55], Thickness[tkns], Line[Map[uG.# &, NuContour /@ νList, {2}]]},
    (* ν = constant curves *)

    {Hue[.15], PointSize[ptsz], Point[MapEyeward[uG.unit[#], .03, eye]] & /@ Λlist},
    (* the eigenvalue triples *)
    {          PointSize[.02],  Point[MapEyeward[uG.unit[#], .02, eye]] & /@ ExtraPts }
   }, ViewPoint → eye, Lighting → "Neutral", Boxed -> False]
```

```
In[91]:= eye = 10. xyztp[{20, 70}]; (* 20 is azimuth of eye, 70 is colatitude of eye *)
         Λlist = {{2, 1, 0}, {2, 1, -3}, {2, 2, -1} }; (* your eigenvalue triples;
         you specify. You can read them in from a data file. They do not have to be normalized,
         but they have to be ordered if you want them on the lune *)
         ExtraPts = {{1, 0, -1}, {1, 1, -2}, {2, -1, -1}}; (* other points,
         in Λ coordinates; you specify. They do not have to be normalized. *)
         uvwAxes = False;
         Λaxes = False;
         grid = False;
         ZeroLam1Lam3 = True;
         DeviatoricArc = True;
         LuneBoundary = True;
         νList = {0, 1 / 4};  (* specifies the Poisson contours;
         make νList = () if you do not want any. *)
         tkns = .006; ptsz = .02; ArrowScale = .055;
         (* ptsz is point size for the eigenvalue triples *)
         Lune[Λlist, ExtraPts, eye, Λaxes, uvwAxes,
          grid, ZeroLam1Lam3, DeviatoricArc, LuneBoundary, νList]
```

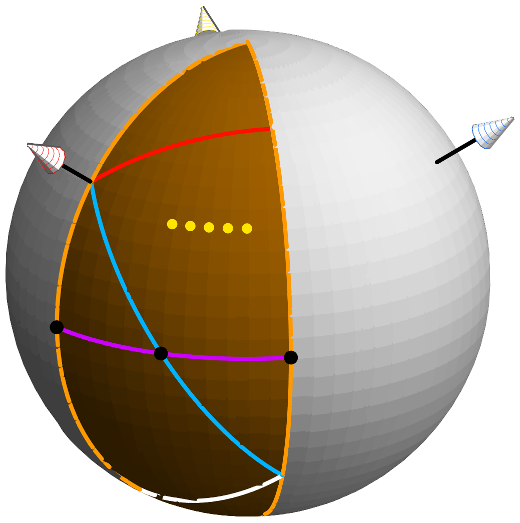
Out[102]=



Now experiment with the inputs :

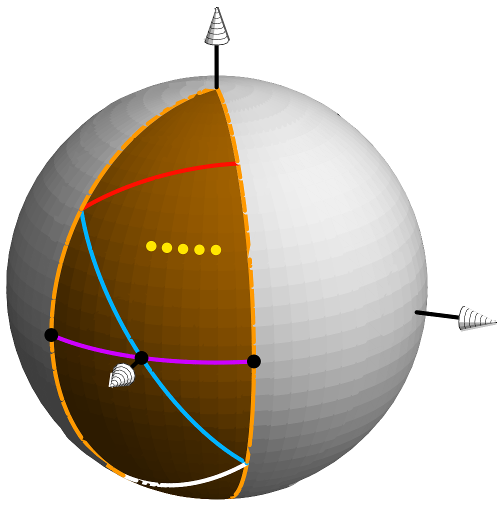
```
In[103]:= eye = 10. xyztp[{20, 70}];
          Λlist = Table[ΛofβΥ[{60 Degree, Υ}], {Υ, 0., 20 Degree, 5 Degree}];
          ExtraPts = {{1, 0, -1}, {1, 1, -2}, {2, -1, -1}};
          uvwAxes = False;
          Λaxes = True;
          grid = False;
          ZeroLam1Lam3 = True;
          DeviatoricArc = True;
          LuneBoundary = True;
          νList = {0};  (* ν = 0 contour is λ2 = 0 *)
          tkns = .006; ptsz = .015; ArrowScale = .055;
          Lune[Λlist, ExtraPts, eye, Λaxes, uvwAxes,
           grid, ZeroLam1Lam3, DeviatoricArc, LuneBoundary, νList]
```

Out[114]=



In[115]:=
```
uvwAxes = True;
Λaxes = False;
Lune[Λlist, ExtraPts, eye, Λaxes, uvwAxes,
 grid, ZeroLam1Lam3, DeviatoricArc, LuneBoundary, νList]
```

Out[117]=



In[118]:=
```
grid = True;
ZeroLam1Lam3 = False;
DeviatoricArc = False;
νList = {};
Lune[Λlist, ExtraPts, eye, Λaxes, uvwAxes,
 grid, ZeroLam1Lam3, DeviatoricArc, LuneBoundary, νList]
```

Out[122]=