Generate elliptical paraboloids

Using the spherical coordinates to express the components x1,x2,x3 of the angular momentum

The general expression for the paraboloid:

$$\mathcal{P}:\ x_2^2 + \mathrm{u}x_3^2 + 2\, v_0\, x_1$$

- the paraboloid points along the x_1- axis
- the value of v_0 can be both **positive** and **negative:** the paraboloid can point into both the negative and the positive direction of x_1
- The equation for the elliptic paraboloid is given in Cartesian form by $\frac{z}{c} = \frac{x^2}{a^2} + \frac{y^2}{b^2}$

General test for paraboloid + a.m. sphere

```
In[23]:= cp3dpara[u_, v_, e_, r_] :=
   ContourPlot3D[y^2 + u z^2 + 2 v * x == e, {x, -r, r}, {y, -r, r}, {z, -r, r}, Mesh → None,
    ContourStyle → {Blue, Opacity[0.4]}, Boxed → False, Axes → False];
 cp3dams[r_, e_, u_, v_] := ContourPlot3D[y^2 + z^2 + x^2 == r^2, {x, -r, r}, {y, -r, r},
    {z, -r, r}, MeshFunctions → {Function[{x, y, z}, Evaluate[y^2 + u z^2 + 2 v * x]]},
    Mesh → {{e}}, ContourStyle → {Red, Specularity[0.2], Opacity[0.8]},
    MeshStyle → {Yellow, Thick}, PlotPoints → 10,
    BoundaryStyle → None, Boxed → False, Axes → False];
 axes[r_] := {Graphics3D[{Thick, Black, Arrow[{{0, 0, 0}, {0, 0, r + 1.5}}],
      Inset[Style["x3", 20, Bold, Italic, FontFamily → "Times New Roman"],
       Scaled[{0, 0, .1}, {0, 0, r + 1.5}]]}, Boxed → False],
    Graphics3D[{Thick, Blue, Arrow[{{0, 0, 0}, {r + 1.5, 0, 0}}],
      Inset[Style["x1", 20, Bold, Italic, FontFamily → "Times New Roman"],
       Scaled[{0.1, 0, 0}, {r + 1.5, 0, 0}]]}],
    Graphics3D[{Thick, Black, Arrow[{{0, 0, 0}, {0, r + 1.5, 0}}],
      Inset[Style["x2", 20, Bold, Italic, FontFamily → "Times New Roman"],
       Scaled[{0, 0.1, 0}, {0, r + 1.5, 0}]]}]};
 intersection[u_, v_, e_, r_] := {axes[r + 0.3 r], cp3dpara[u, v, e, r + 2],
    cp3dams[r, e, u, v]};
```

```
(*Manipulate[Show[intersection[u,v,e,r],Epilog→
   Inset[Framed[Style[TemplateApply[StringTemplate["I=``\nE=``"][spinvalue,e]],
      15,Bold,Italic,FontFamily→"Times New Roman"],
     Background→None,FrameStyle→None],Scaled[{0.2,0.2},{0.6,0.6}]]],
 {e,0,22,2},{u,-4,4,0.5},{v,-4,4,0.5},{r,2,10,2}]*)
```

Generate proper (u,v) functions

The Hamiltonian of the system (**even-even core + odd particle**) is quadratic in the (x_2, x_3) coordinates, linear in the $x_1$coordinate and it is also dependent on two scale parameters:

$$u = \frac{A_3 - A_1}{A}$$

$$v_0 = v = \frac{-A_1 j_1}{A}$$

The parameter ***A*** that enters the above equations is also defined in terms of the rotational parameters (inverse of **inertia moments** of the nucleus) and the odd-particle angular momentum and coupling angle.

$$A = A_2\left(1 - \frac{j_2}{I}\right) - A_1$$

Based on the coupling angle θ of the odd particle's with the xOy plane, and the angular momentum j of the nucleon, it is possible to construct the scale parameters ***(u,v)*** for an arbitrary value of the total angular momentum ***I***.

The moments of inertia also need to be fixed on order to have definite values for the scale parameters.

Rotational parameters can have different orderings, but the main ones are the ones where each of the three moments of inertia has the biggest value.

```
In[28]:= A[A1_, A2_, I_, j_, θ_] := A2 (1 - j * Sin[θ] / I) - A1;
u[A1_, A2_, A3_, I_, j_, θ_] := (A3 - A1) / A[A1, A2, I, j, θ];
v[A1_, A2_, I_, j_, θ_] := (-A1 * j * Cos[θ]) / A[A1, A2, I, j, θ];
```

Constants

```
In[31]:= j = 5.5;
spinvalue = 19 / 2;
```

- $A_1 < A_2 < A_3$

This is the ordering in which the **moment of inertia along 1-axis is the largest**

```
In[49]:= A11 = 0.0079; A21 = 0.0152; A31 = 0.0385; theta1 = -1.95;
```

- $A_2 < A_3 < A_1$

```
In[60]:= A12 = 0.0417; A22 = 0.0093; A32 = 0.0278; theta2 = -1.71;
```

This is the ordering in which the **moment of inertia along 2-axis is the largest**

- $A_3 < A_2 < A_1$

In[63]:=
```
A13 = 0.0357; A23 = 0.0625; A33 = 0.0104; theta3 = 1.64;
```

This is the ordering in which the **moment of inertia along 3-axis is the largest**

In[36]:=
```
genUVparams[I_, k_] :=
  { {u[A11, A21, A31, I, j, theta1], v[A11, A21, I, j, theta1]}  k == 1
    {u[A12, A22, A32, I, j, theta2], v[A12, A22, I, j, theta2]}  k == 2 ;
    {u[A13, A23, A33, I, j, theta3], v[A13, A23, I, j, theta3]}  k == 3
```

In[37]:=
```
uv[spin_] := {genUVparams[spin, 1], genUVparams[spin, 2], genUVparams[spin, 3]};
```

In[38]:=
```
quantizationPlot[r_, e_, k_] := {intersection[uv[r][[k, 1]], uv[r][[k, 2]], e, r]};
```

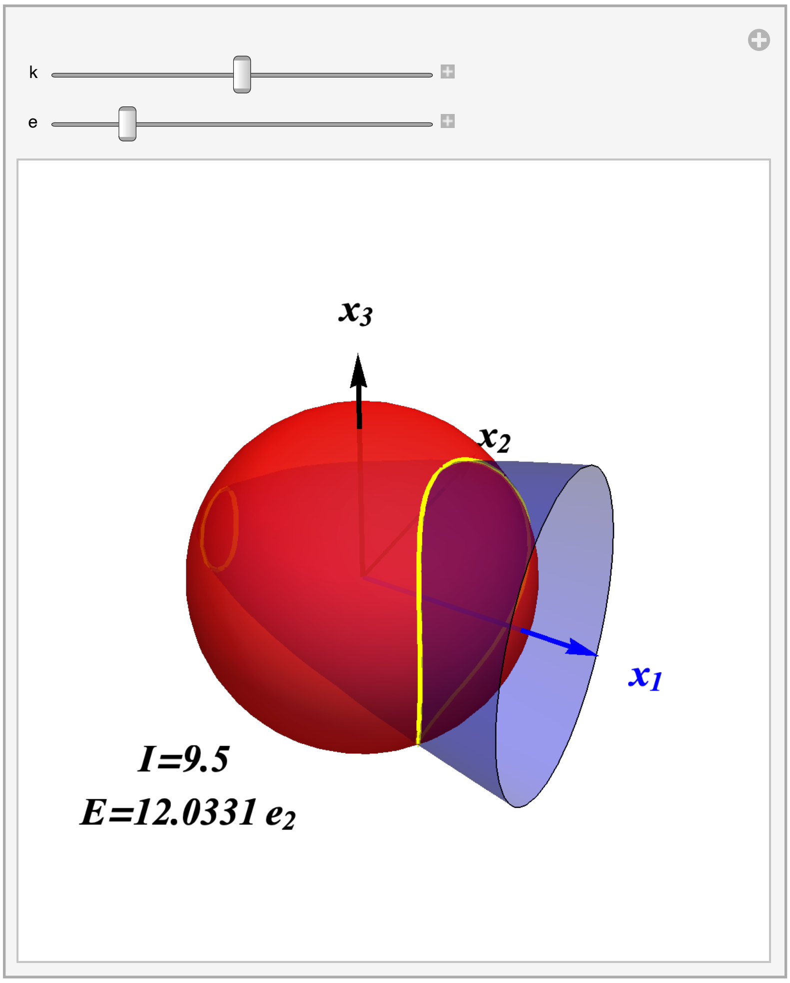
Energy units for each axis quantisations

In[90]:=
```
ek = {1.0558, 2.0776, 0.9386};
```

Find the critical spin value

In[99]:=
```
Manipulate[Show[quantizationPlot[spinvalue, e, k],
  Epilog → Inset[Framed[Style[TemplateApply[
      StringTemplate["I=``\n E=``e``"][spinvalue, N[e/ek[[k]], 2], k]], 20,
     Bold, Italic, Black, FontFamily → "Times New Roman"], FrameStyle → None],
    Scaled[{0.2, 0.2}]]], {k, 1, 3, 1}, {e, 0, 150, 1}]
```

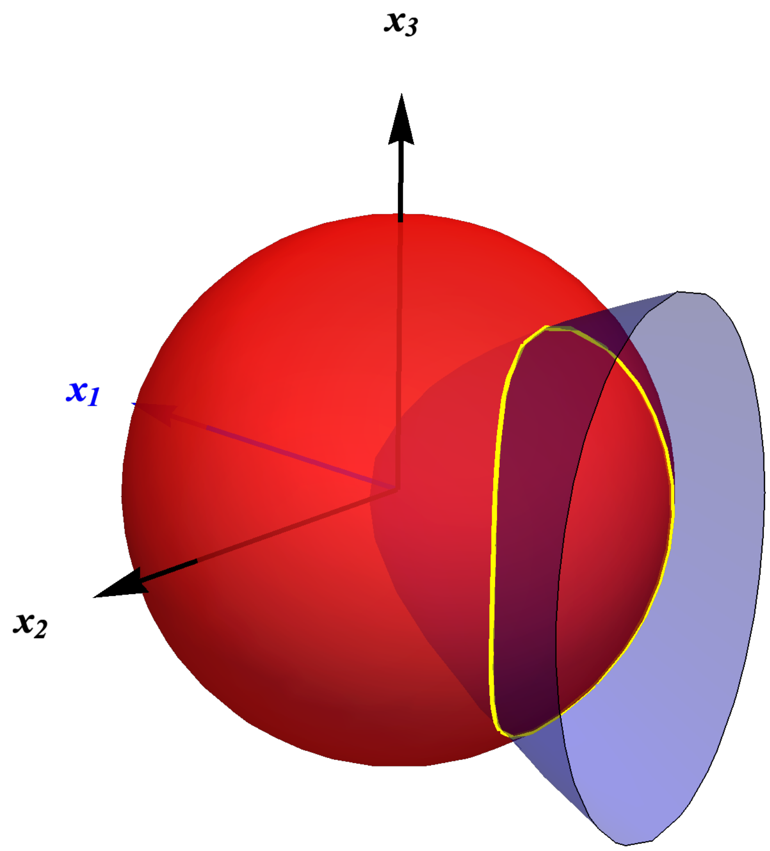
Out[99]=



Function to get the view point of the plot

In[40]:=
```
Get3DView[gfx_] :=
  Options[Unevaluated[gfx], {ViewCenter, ViewVector, ViewVertical, ViewPoint}];
```

In[41]:= plotpar = Get3DView[

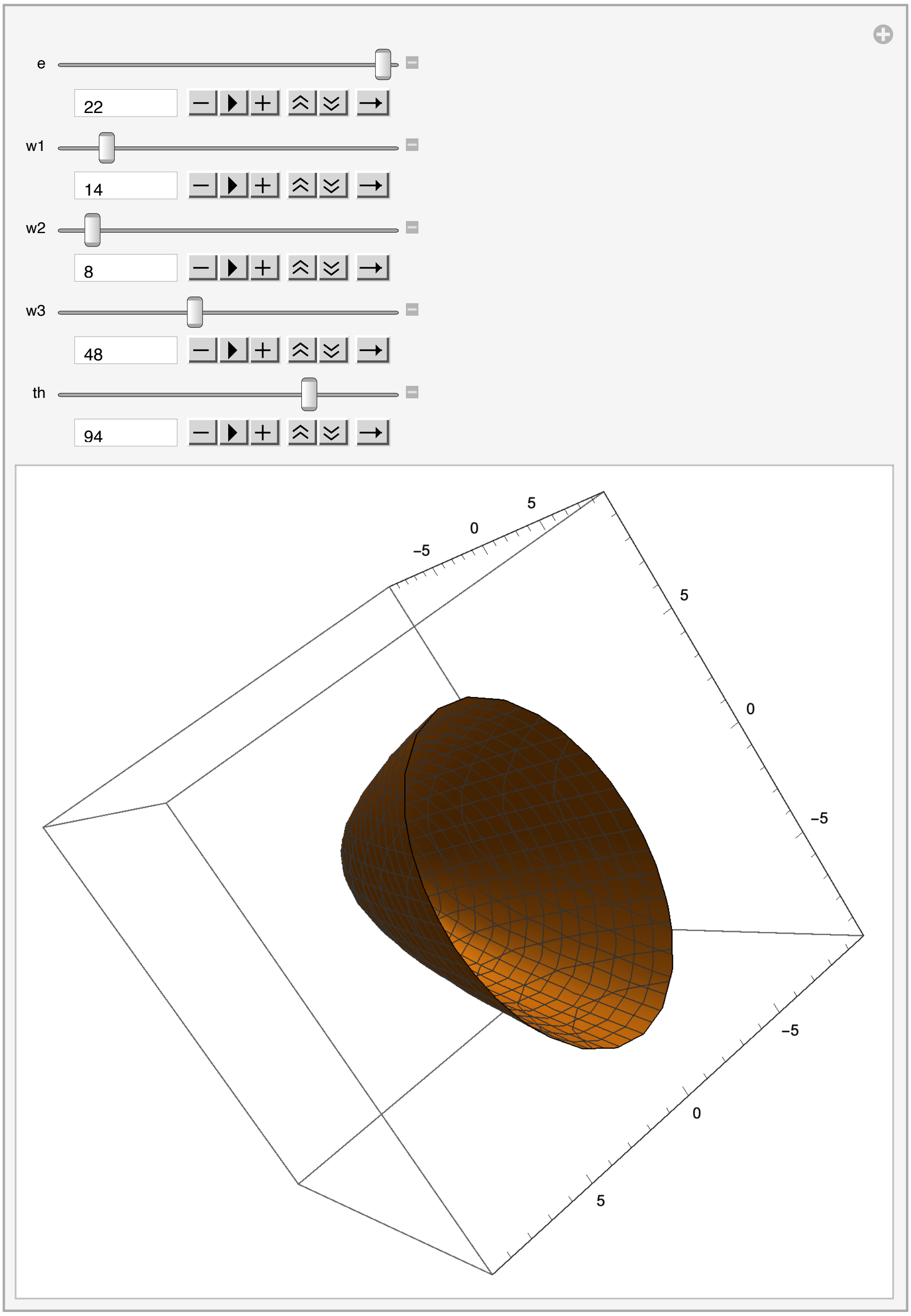


];

Testing the paraboloid in terms of scale parameters (u, v)

In[48]:=
```
Manipulate[ContourPlot3D[y^2 + u[1/(2*w1), 1/(2*w2), 1/(2*w3), 9.5, j, th*π/180]*z^2 +
   2*v[1/(2*w1), 1/(2*w2), 9.5, j, th*π/180]*x == e, {x, -spinvalue, spinvalue},
  {y, -spinvalue, spinvalue}, {z, -spinvalue, spinvalue}], {e, 0, 22, 2},
 {w1, 1, 120, 1}, {w2, 1, 120, 1}, {w3, 1, 120, 1}, {th, -180, 180, 1}]
```

Out[48]=



Power: Infinite expression $\frac{1}{0.}$ encountered.

Infinity: Indeterminate expression 0 ComplexInfinity encountered.

Power: Infinite expression $\frac{1}{0.}$ encountered.

Power: Infinite expression $\frac{1}{0.}$ encountered.

General: Further output of Power::infy will be suppressed during this calculation.

Infinity: Indeterminate expression 0 ComplexInfinity encountered.

Infinity: Indeterminate expression 0 ComplexInfinity encountered.

General: Further output of Infinity::indet will be suppressed during this calculation.

Power: Infinite expression $\frac{1}{0.}$ encountered.

Power: Infinite expression $\frac{1}{0.}$ encountered.

Infinity: Indeterminate expression y^2 + ComplexInfinity + ComplexInfinity encountered.

Power: Infinite expression $\frac{1}{0.}$ encountered.

General: Further output of Power::infy will be suppressed during this calculation.

Infinity: Indeterminate expression y^2 + ComplexInfinity + ComplexInfinity encountered.

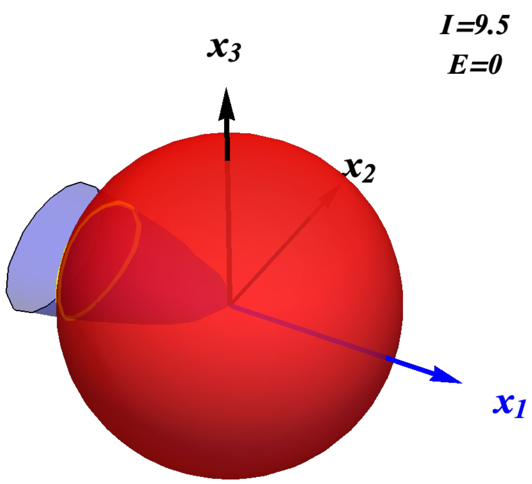
Infinity: Indeterminate expression 90.2242 + ComplexInfinity + ComplexInfinity encountered.

General: Further output of Infinity::indet will be suppressed during this calculation.

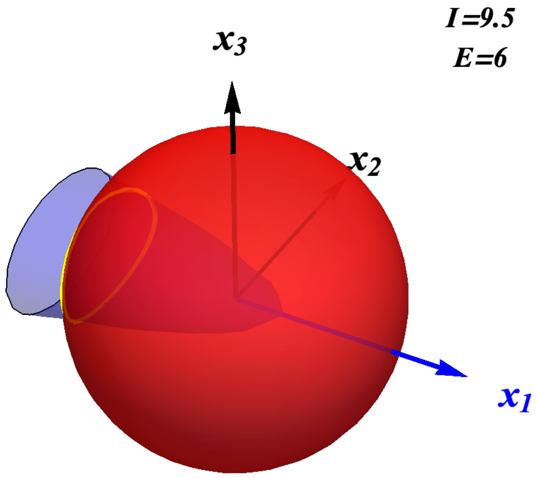
All plots

1-AXIS QUANTIZATION (1-AXIS has the maximal MOI)

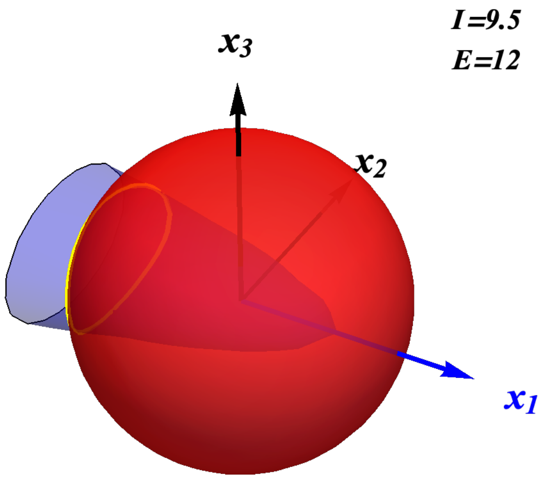
In[78]:=
```
Table[Show[quantizationPlot[spinvalue, e, 1],
  Epilog → Inset[Framed[Style[TemplateApply[
       StringTemplate["I=``\nE=``"][spinvalue, e]], 15, Bold, Italic,
      FontFamily → "Times New Roman"], Background → None, FrameStyle → None],
    Scaled[{0.2, 0.2}, {0.6, 0.6}]], ImageSize -> Medium], {e, 0, 30, 6}]
```

{ 

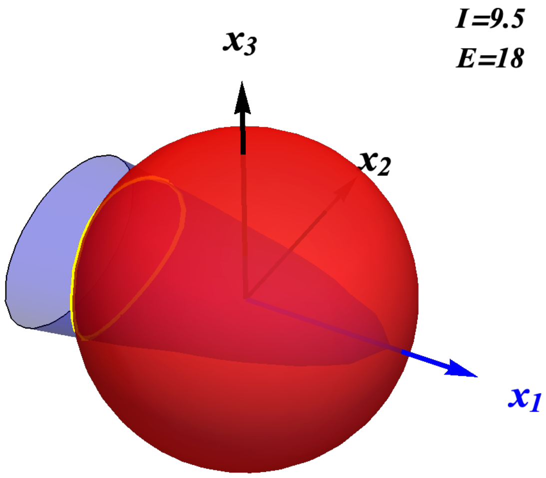
,


I=9.5
E=6
x3
x2
x1

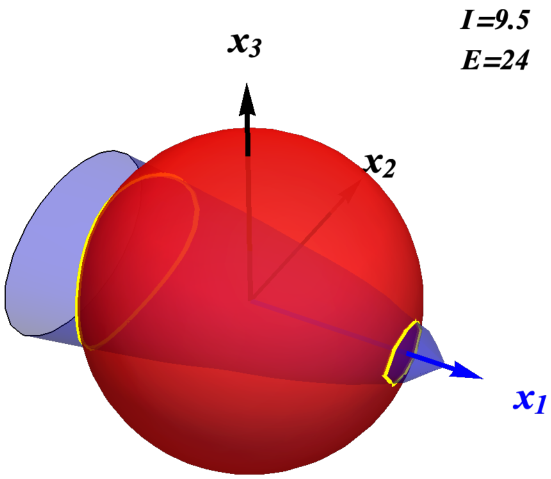
,


I=9.5
E=12
x3
x2
x1

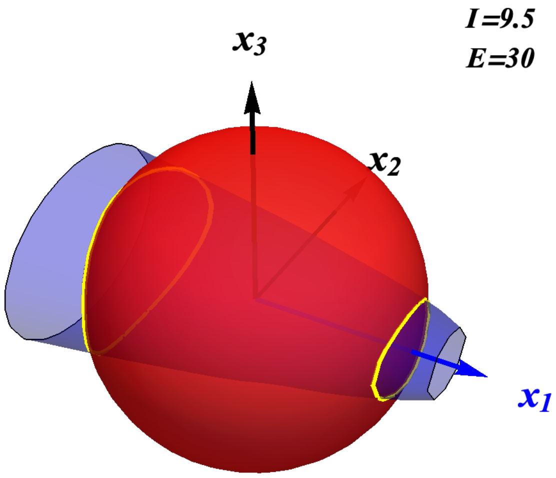
,


I=9.5
E=18
x3
x2
x1

,


I=9.5
E=24
x3
x2
x1

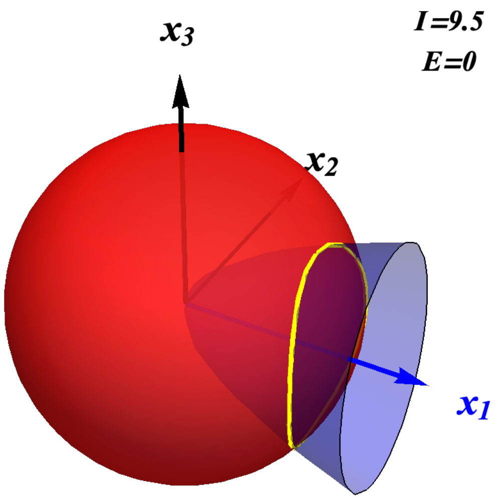
,

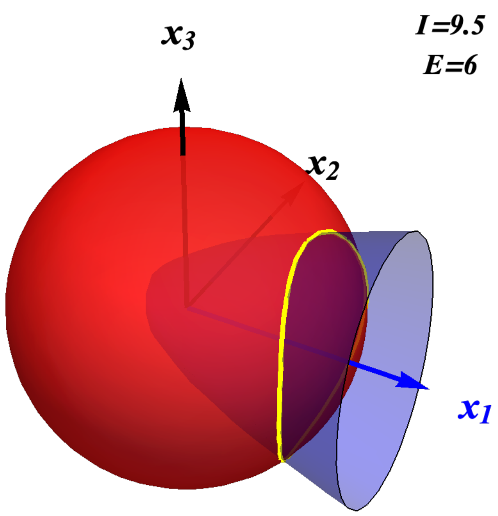


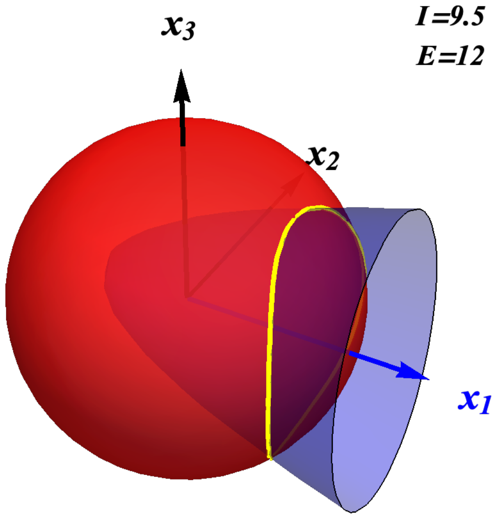
}

2-AXIS QUANTIZATION (1-AXIS has the maximal MOI)

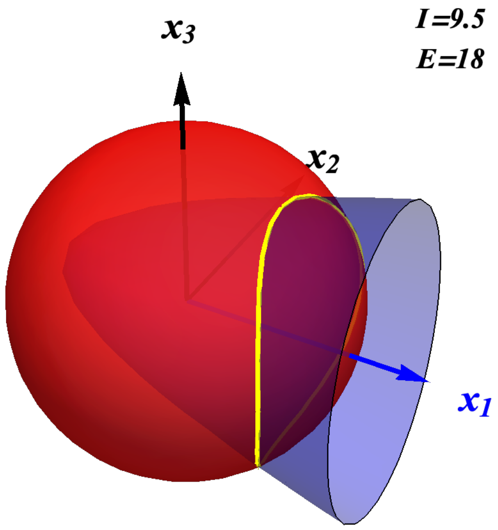
In[69]:=
```
Table[Show[quantizationPlot[spinvalue, e, 2],
  Epilog → Inset[Framed[Style[TemplateApply[
      StringTemplate["I=``\nE=``"][spinvalue, e]], 15, Bold, Italic,
     FontFamily → "Times New Roman"], Background → None, FrameStyle → None],
   Scaled[{0.2, 0.2}, {0.6, 0.6}]], ImageSize -> Medium], {e, 0, 30, 6}]
```

{ 
I=9.5
E=0
x3
x2
x1
,

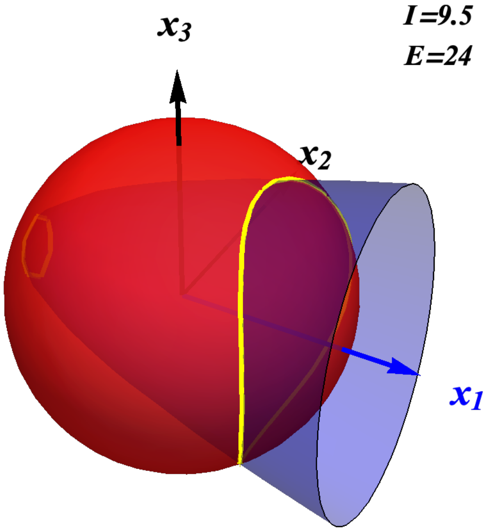

I=9.5
E=6
x3
x2
x1
,


x_3
I=9.5
E=12
x_2
x_1

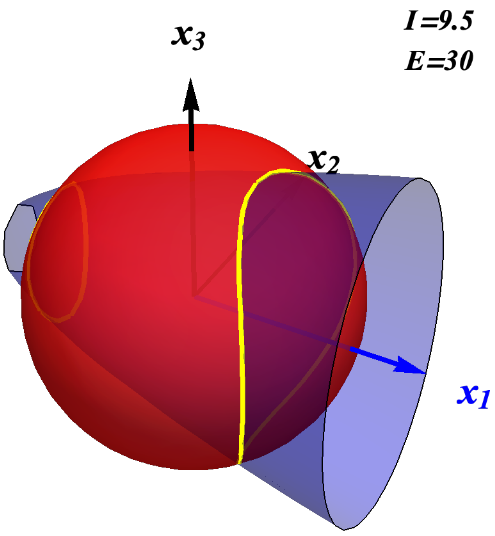
,


x_3
I=9.5
E=18
x_2
x_1

,



,

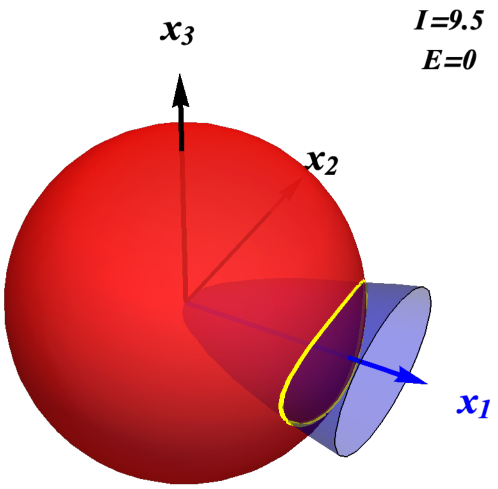


}

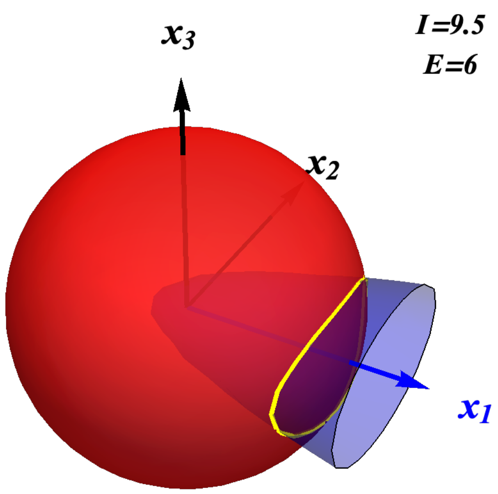
3-AXIS QUANTIZATION (1-AXIS has the maximal MOI)

In[70]:=
```
Table[Show[quantizationPlot[spinvalue, e, 3],
  Epilog → Inset[Framed[Style[TemplateApply[
        StringTemplate["I=``\nE=``"][spinvalue, e]], 15, Bold, Italic,
      FontFamily → "Times New Roman"], Background → None, FrameStyle → None],
    Scaled[{0.2, 0.2}, {0.6, 0.6}]], ImageSize -> Medium], {e, 0, 30, 6}]
```

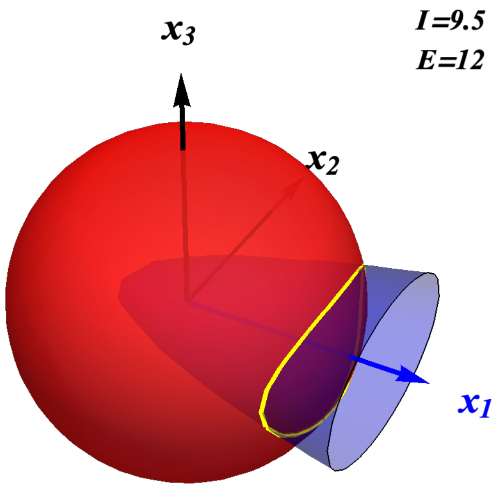
Out[70]= {



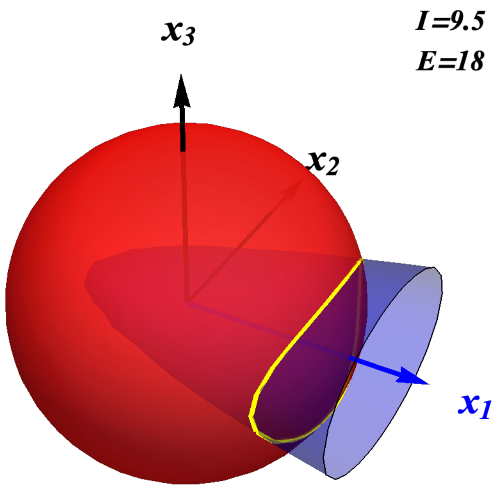
,



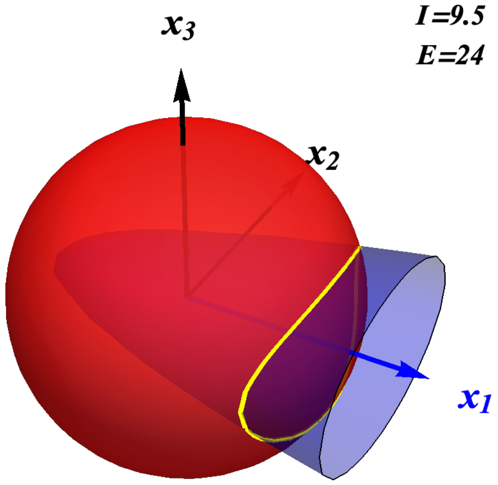
,


I=9.5
E=12
x_3
x_2
x_1

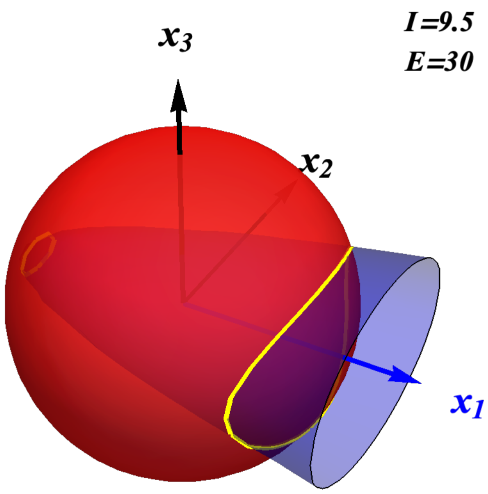
,


I=9.5
E=18
x_3
x_2
x_1

,


x_3
I=9.5
E=24
x_2
x_1

,


x_3
I=9.5
E=30
x_2
x_1

}