Some 3D orthogonal coordinate systems

PHYS/MATH 4325
Texas Tech University
Prof. Katharine Long

The most important 3D orthogonal coordinate systems are Cartesian, cylindrical, and spherical; you need to learn to work comfortably with all three.

Others are occasionally useful. For example, several times over the years I’ve found use for oblate spheroidal coordinates in engineering applications. A very general system, the triaxial ellipsoidal coordinate system (not shown here) has been used for some interesting problems in theoretical astrophysics (see, for example, *Galactic Dynamics* by J. Binney and S. Tremaine).

A WORD ABOUT THE GRAPHICS: You can manipulate the figures without re-evaluating the notebook. If you do choose to re-evaluate the notebook, it might take a few minutes depending on the speed of your computer.

```
In[∘]:= plotCoords3D[R_, {u_, uMin_, uMax_, nu_ : 4},
          {v_, vMin_, vMax_, nv_ : 4}, {w_, wMin_, wMax_, nw_ : 4}, args___] :=
         Show[
          ParametricPlot3D[Table[R[u, v, w], {u, uMin, uMax, (uMax - uMin) / nu}],
           {v, vMin, vMax}, {w, wMin, wMax},
           PlotStyle → Directive[Blue, Opacity[0.2]], Mesh → None, args],
          ParametricPlot3D[Table[R[u, v, w], {v, vMin, vMax, (vMax - vMin) / nv}],
           {u, uMin, uMax}, {w, wMin, wMax},
           PlotStyle → Directive[Red, Opacity[0.2]], Mesh → None, args],
          ParametricPlot3D[Table[R[u, v, w], {w, wMin, wMax, (wMax - wMin) / nw}],
           {u, uMin, uMax}, {v, vMin, vMax},
           PlotStyle → Directive[Green, Opacity[0.2]], Mesh → None, args],
          AxesOrigin → {0, 0, 0},
          AxesLabel → {x, y, z},
          Boxed → False,
          ViewVector → {3, 8, 4},
          ViewVertical → {0, 0, 1},
          AxesStyle → Directive[Thick, Medium],
          args
         ]
```

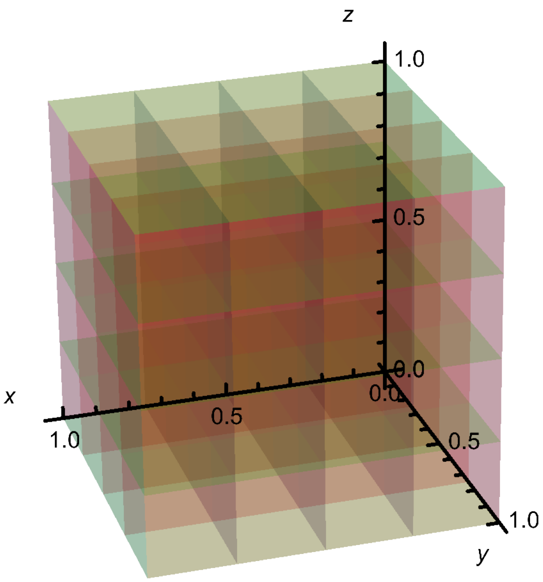
- ## Cartesian coordinates

```
In[∘]:= rcart3D[x_, y_, z_] = {x, y, z}
```

Out[∘]= $\{x, y, z\}$

In[•]:=
```
plotCoords3D[r_cart3D, {x, 0, 1}, {y, 0, 1}, {z, 0, 1}]
```

Out[•]=



Cylindrical coordinates

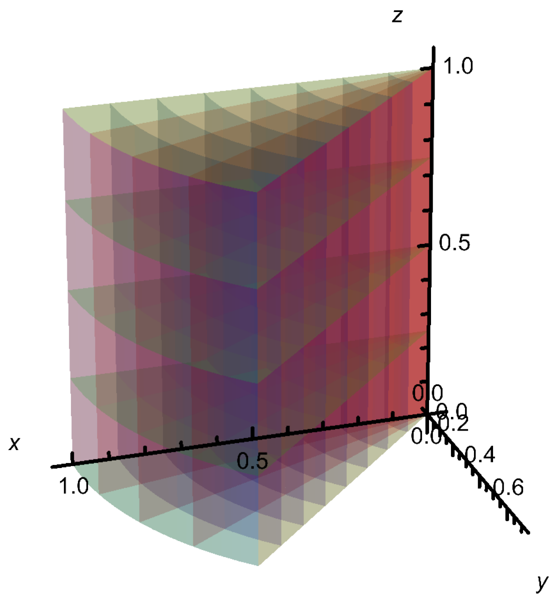
Many engineering and biomedical applications: flow in pipes and blood vessels, ion channels in cell membranes, structural members from flagpoles to bones, reactor cooling rods, wires, many more.

In[•]:=
```
r_cyl[ρ_, ϕ_, z_] = {ρ Cos[ϕ], ρ Sin[ϕ], z}
```

Out[•]= $\{\rho\cos(\phi), \rho\sin(\phi), z\}$

In[•]:=
```
plotCoords3D[rcyl, {ρ, 0, 1, 8}, {ϕ, 0, Pi / 4}, {z, 0, 1}]
```

Out[•]=



- ## Elliptic cylindrical coordinates

See Hassani problem 2.35

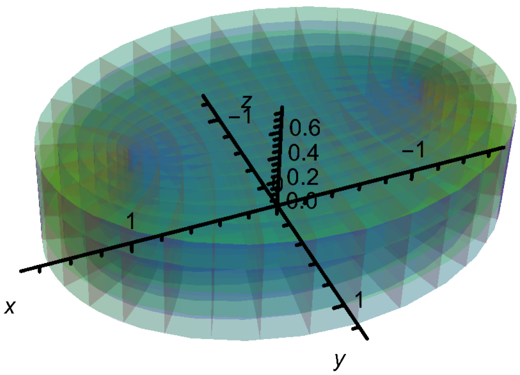
Applications: I don’t know of any useful applications. If you ever run into one, please let me know.

In[•]:=
```
rellCyl[u_, v_, z_] = {Cosh[u] Cos[v], Sinh[u] Sin[v], z}
```

Out[•]= $\{\cosh(u)\cos(v), \sinh(u)\sin(v), z\}$

```
In[∘]:= plotCoords3D[rellCyl, {u, -1, 1, 16}, {ϕ, 0, Pi, 16}, {z, 0, 3/4},
          ViewVector → {2, 8, 12}]
```

Out[∘]=



- ## Spherical coordinates

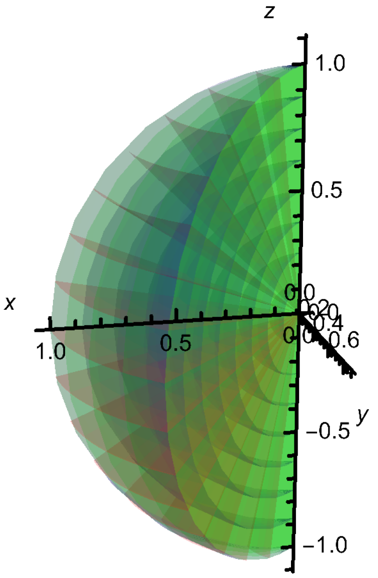
  Applications: anything round, or approximately so, from atoms to galaxies

```
In[∘]:= rsph[ρ_, θ_, ϕ_] = ρ {Sin[θ] Cos[ϕ], Sin[θ] Sin[ϕ], Cos[θ]}
```

Out[∘]= $\{\rho \sin(\theta) \cos(\phi), \rho \sin(\theta) \sin(\phi), \rho \cos(\theta)\}$

```
In[•]:= plotCoords3D[r_sph, {ρ, 0, 1, 8}, {θ, 0, Pi, 16}, {ϕ, 0, Pi/4}]
```

Out[•]=



• Prolate spheroidal coordinates

See Hassani, problem 2.36

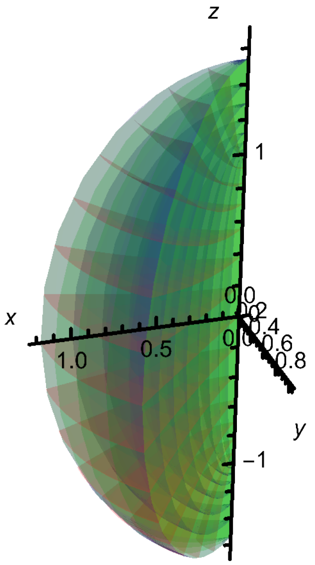
Applications: gravitational or electrostatic fields of stretched spheroids (e.g, elliptical galaxies, Hyperion), scattering of waves from ellipsoidal dust grains, dynamics of grains in liquid crystals, fluid flow around elongated bacteria (for example, *E. coli* or *Y. pestis*)

```
In[•]:= r_prol[u_, θ_, ϕ_] = {
          Sinh[u] Sin[θ] Cos[ϕ],
          Sinh[u] Sin[θ] Sin[ϕ],
          Cosh[u] Cos[θ]
        }
```

Out[•]= $\{\sin(\theta)\sinh(u)\cos(\phi), \sin(\theta)\sinh(u)\sin(\phi), \cos(\theta)\cosh(u)\}$

```
In[•]:= plotCoords3D[rprol, {u, 0, 1, 8}, {θ, 0, Pi, 16}, {ϕ, 0, Pi / 4}]
```

Out[•]=



- ## Oblate spheroidal coordinates

See, e.g, Arfken and Weber or Morse and Feschbach. For some reason this one doesn’t seem to be in Hassani

This one is more useful than you might guess; I’ve used it in several engineering applications.

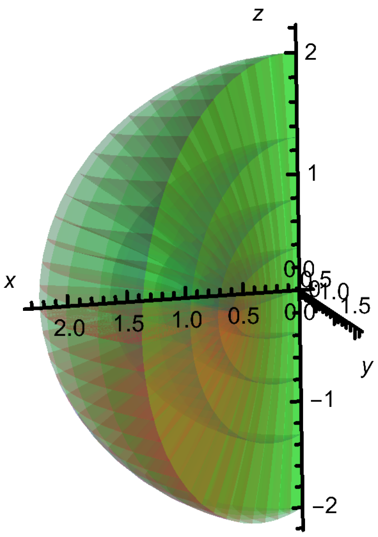
Some applications: gravitational or electrostatic fields of flattened spheroids (e.g, Earth, Sun, elliptical galaxies), scattering of waves from ellipsoidal dust grains, dynamics of grains in liquid crystals, water flow around flattened cells (for example, red blood cells), elastic deformation of optical lenses, ideal flow out of a circular hole (such as an ion channel in a cell membrane), electrostatic potential above a plate with a circular hole, magnetic field near the end of a cylindrical solenoid.

```
In[•]:= robl[u_, θ_, ϕ_] = {Cosh[u] Cos[θ] Cos[ϕ], Cosh[u] Cos[θ] Sin[ϕ], Sinh[u] Sin[θ]}
```

Out[•]= $\{\cos(\theta)\cosh(u)\cos(\phi),\ \cos(\theta)\cosh(u)\sin(\phi),\ \sin(\theta)\sinh(u)\}$

In[•]:=
```
plotCoords3D[r_obl, {u, -ArcSinh[2], ArcSinh[2], 8}, {θ, 0, Pi/2, 16}, {ϕ, 0, Pi/4}]
```

Out[•]=



- ## Paraboloidal coordinates

See Hassani, prob 2.38

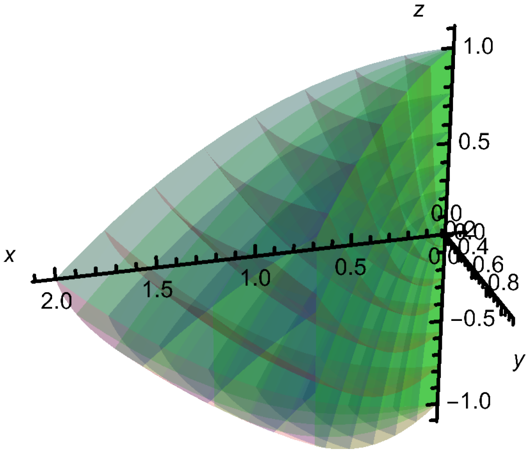
Application: parabolic antennas

In[•]:=
```
r_par3D[u_, v_, ϕ_] = {2 u v Cos[ϕ], 2 u v Sin[ϕ], u^2 - v^2}
```

Out[•]= $\{2\,u\,v\cos(\phi),\ 2\,u\,v\sin(\phi),\ u^2 - v^2\}$

In[•]:=
```
plotCoords3D[r_par3D, {u, 0, 1, 8}, {v, 0, 1, 8}, {ϕ, 0, Pi / 4}]
```

Out[•]=



- ## Toroidal coordinates

WARNING: the definition of toroidal coordinates given in problem 2.37 of Hassani has several typographical errors. You can find correct expressions online (Wikipedia or Wolfram MathWorld) or in textbooks such as Arfken and Weber or Morse and Feschbach.

Applications: plasma confinement, vortex rings, magnetic fields around toroidal solenoids

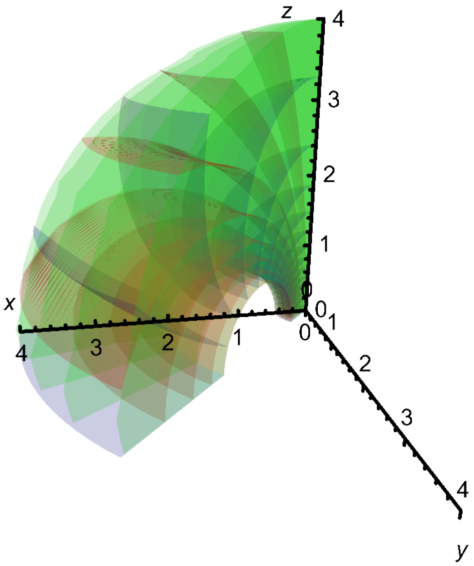
In[•]:=
```
r_tor[σ_, τ_, ϕ_] = {Sinh[τ] Cos[ϕ], Sinh[τ] Sin[ϕ], Sin[σ]} / (Cosh[τ] - Cos[σ])
```

Out[•]=

$$\left\{\frac{\sinh(\tau)\cos(\phi)}{\cosh(\tau)-\cos(\sigma)}, \frac{\sinh(\tau)\sin(\phi)}{\cosh(\tau)-\cos(\sigma)}, \frac{\sin(\sigma)}{\cosh(\tau)-\cos(\sigma)}\right\}$$

In[∘]:=
```
plotCoords3D[rtor, {σ, 0, Pi, 16}, {τ, 0, 1, 8}, {ϕ, 0, Pi/4},
  PlotRange → {{0, 4}, {0, 4}, {0, 4}},
      RegionFunction → Function[{x, y, z}, x^2 + y^2 + z^2 ≤ 16]]
```

Out[∘]=



- ## Bipolar coordinates

See Hassani, prob 2.39

Applications: electronic structure of diatomic molecules in the Born-Oppenheimer approximation, electrostatic fields due to two spheres.

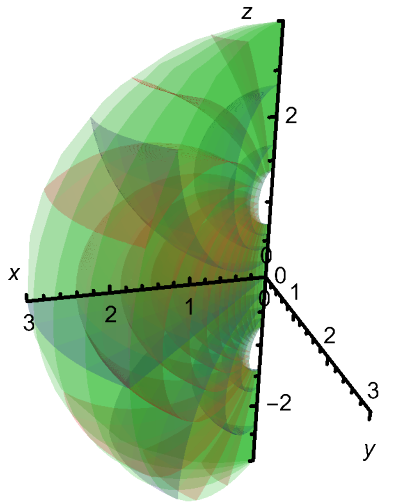
In[∘]:=
```
rbip[u_, θ_, ϕ_] = {Sin[θ] Cos[ϕ], Sin[θ] Sin[ϕ], Sinh[u]} / (Cosh[u] - Cos[θ])
```

Out[∘]=

$$\left\{\frac{\sin(\theta)\cos(\phi)}{\cosh(u)-\cos(\theta)}, \frac{\sin(\theta)\sin(\phi)}{\cosh(u)-\cos(\theta)}, \frac{\sinh(u)}{\cosh(u)-\cos(\theta)}\right\}$$

In[•]:=
```
plotCoords3D[rbip, {u, -ArcSinh[3], ArcSinh[3], 8}, {θ, 0, Pi, 16}, {ϕ, 0, Pi / 4},
 PlotRange → {{0, 3}, {0, 3}, {-3, 3}},
    RegionFunction → Function[{x, y, z}, x^2 + y^2 + z^2 ≤ 9]]
```

Out[•]=


z
2
0
x
3
2
1
0
0
1
2
3
−2
y