Well Geometry

Robert Atkinson
27 Sept 2019

We explore the geometry of various labware.

Administrivia

Basics

Cone

Accessing

```
assumptions[cone[h_, r_]] := h >= 0 && r >= 0
assumptions[cone[h_, α_, apexangle]] := FullSimplify[h >= 0 && α > 0 && α < π / 2]
assumptions[cone[h_, β_, baseangle]] := FullSimplify[assumptions[cone[h, complement[β], apexangle]]]
```

```
test @ assumptions[cone[h, α, apexangle]];
test @ assumptions[cone[h, β, baseangle]];
```

```
assumptions[cone[h, α, apexangle]] → h ≥ 0 && α > 0 && 2 α < π
```

```
assumptions[cone[h, β, baseangle]] → h ≥ 0 && 2 β < π && β > 0
```

```
radius[c : cone[h_, r_]] := r
radius[c : cone[h_, α_, apexangle]] := h Tan[α]
radius[c : cone[h_, β_, baseangle]] := h Cot[β]

height[c : cone[h_, r_]] := h
height[c : cone[h_, α_, apexangle]] := h
height[c : cone[h_, β_, baseangle]] := h
```

```
apexangle[c : cone[h_, r_]] := Assuming[assumptions[c], ArcTan[h, r]]
apexangle[c : cone[h_, α_, apexangle]] := α
apexangle[c : cone[h_, β_, baseangle]] := complement[baseangle[c]]
baseangle[c : cone[h_, r_]] := Assuming[assumptions[c], ArcTan[r, h]]
baseangle[c : cone[h_, α_, apexangle]] := complement[α]
baseangle[c : cone[h_, β_, baseangle]] := β
```

```
test @ apexangle[cone[h, r]];
test @ apexangle[cone[h, α, apexangle]];
test @ apexangle[cone[h, β, baseangle]];
test @ baseangle[cone[h, r]];
test @ baseangle[cone[h, α, apexangle]];
test @ baseangle[cone[h, β, baseangle]];
```

```
apexangle[cone[h, r]] → ArcTan[h, r]
```

```
apexangle[cone[h, α, apexangle]] → α
```

apexangle[cone[h, β, baseangle]] → $\frac{\pi}{2} - \beta$

```
baseangle[cone[h, r]] → ArcTan[r, h]
```

baseangle[cone[h, α, apexangle]] → $\frac{\pi}{2} - \alpha$

```
baseangle[cone[h, β, baseangle]] → β
```

Conversion

Volume

Height and Depth

Inverted Cone

Conversion

Accessing

```
assumptions[c : invertedCone[h_, r_]] := assumptions[toCone @ c]
assumptions[c : invertedCone[h_, α_, apexangle]] := assumptions[toCone @ c]
assumptions[c : invertedCone[h_, β_, baseangle]] := assumptions[toCone @ c]
test @ assumptions[invertedCone[h, α, apexangle]];
test @ assumptions[invertedCone[h, β, baseangle]];
```

```
assumptions[invertedCone[h, α, apexangle]] → h ≥ 0&&α > 0&&2 α < π
```

```
assumptions[invertedCone[h, β, baseangle]] → h ≥ 0&&2 β < π&&β > 0
```

```
radius[c : invertedCone[h_, r_]] := r
radius[c : invertedCone[h_, α_, apexangle]] := radius @ invert @ c
radius[c : invertedCone[h_, β_, baseangle]] := radius @ invert @ c

height[c : invertedCone[h_, r_]] := h
height[c : invertedCone[h_, α_, apexangle]] := h
height[c : invertedCone[h_, β_, baseangle]] := h
```

```
apexangle[c : invertedCone[h_, r_]] := Assuming[assumptions[c], ArcTan[h, r]]
apexangle[c : invertedCone[h_, α_, apexangle]] := α
apexangle[c : invertedCone[h_, β_, baseangle]] := complement[baseangle[c]]
baseangle[c : invertedCone[h_, r_]] := Assuming[assumptions[c], ArcTan[r, h]]
baseangle[c : invertedCone[h_, α_, apexangle]] := complement[α]
baseangle[c : invertedCone[h_, β_, baseangle]] := β
```

```
test @ apexangle[invertedCone[h, r]];
test @ apexangle[invertedCone[h, α, apexangle]];
test @ apexangle[invertedCone[h, β, baseangle]];
test @ baseangle[invertedCone[h, r]];
test @ baseangle[invertedCone[h, α, apexangle]];
test @ baseangle[invertedCone[h, β, baseangle]];
```

apexangle[invertedCone[h, r]] → ArcTan[h, r]

apexangle[invertedCone[h, α, apexangle]] → α

apexangle[invertedCone[h, β, baseangle]] → $\frac{\pi}{2} - \beta$

baseangle[invertedCone[h, r]] → ArcTan[r, h]

baseangle[invertedCone[h, α, apexangle]] → $\frac{\pi}{2} - \alpha$

baseangle[invertedCone[h, β, baseangle]] → β

Conversion Redux

Volume

Height and Depth

Final

```
genericInvertedConeDepthFromVolume[] := Module[{c, h, α, hh, vol, a, eqn, solns, soln},
  c = invertedCone[h, α, apexangle];
  a = assumptions[c] && vol ≥ 0;
  eqn = FullSimplify[vol == volume[c], a];
  solns = Assuming[a, Solve[eqn, h]];
  soln = FullSimplify[h /. solns[[2]], a];
  genericInvertedConeDepthFromVolume[] = {α, vol, soln}
 ]
test @ genericInvertedConeDepthFromVolume[];
```

genericInvertedConeDepthFromVolume[] → {α\$64755, vol\$64755, $\left(\frac{3}{\pi}\right)^{1/3} \left(\text{vol\$64755 Cot}[\alpha\$64755]^2\right)^{1/3}$}

```
depthFromVolume[c : invertedCone[ignored_, α_, apexangle], v_] := Module[{αα, vol, soln},
  {αα, vol, soln} = genericInvertedConeDepthFromVolume[];
  (soln /. {αα → α, vol → v}) // FullSimplify
 ]
depthFromVolume[c : invertedCone[h_, r_], v_] := depthFromVolume[toApexAngled @ c, v]
depthFromVolume[c : invertedCone[h_, β_, baseangle], v_] := depthFromVolume[toApexAngled @ c, v]

test @ depthFromVolume[invertedCone[ignored, α, apexangle], volume];
test @ depthFromVolume[invertedCone[h, r], volume];
test @ depthFromVolume[invertedCone[h, β, baseangle], volume];
```

depthFromVolume[invertedCone[ignored, α, apexangle], volume] → $\left(\frac{3}{\pi}\right)^{1/3} \left(\text{volume Cot}[\alpha]^2\right)^{1/3}$

depthFromVolume[invertedCone[h, r], volume] → $\left(\frac{3}{\pi}\right)^{1/3} \left(\frac{h^2\ \text{volume}}{r^2}\right)^{1/3}$

depthFromVolume[invertedCone[h, β, baseangle], volume] → $\left(\frac{3}{\pi}\right)^{1/3} \left(\text{volume Tan}[\beta]^2\right)^{1/3}$

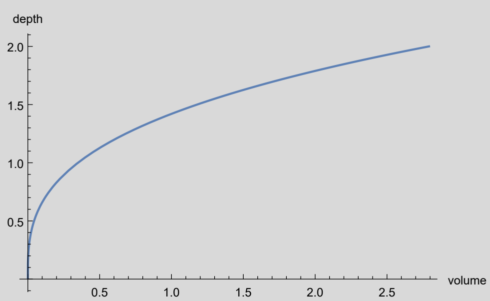
Testing

```
example = invertedCone[2, π / 6, apexangle]
{ volume[example], volume[example] // N }
expr = test @ depthFromVolume[example, v];
Plot[expr, {v, 0, volume[example]}, AxesLabel → {"volume", "depth"}]
```

invertedCone$\left[2, \frac{\pi}{6}, \text{apexangle}\right]$

$\left\{\frac{8\pi}{9}, 2.79253\right\}$

depthFromVolume[example, v] → $\frac{3^{2/3}\ v^{1/3}}{\pi^{1/3}}$



Cylinder

Right Conical Frustum

Inverted Right Conical Frustum

Sphere

Inverted Spherical Cap

Conical Test Tube

Our model of a conical test tube is an “cylindrical” inverted frustum on top of a “conical” inverted frustum on top of an inverted spherical cap

Accessing

```
toCanonical[c : conicalTestTube[cylindrical_, conical_, cap_]] := c
toCanonical[conicalTestTube[{idTop_, idHip_, idBottom_}, {hTop_, hBottomAndCap_}]] := conicalTestTube[
  (* TODO: use cylinders when we need to *)
  invertedFrustum[hTop, idTop / 2 , idHip / 2 ],
  invertedFrustum[hBottomAndCap - idBottom, idHip / 2 , idBottom / 2 ],
  cap = invertedSphericalCap[idBottom / 2, idBottom / 2 ]
 ]
```

```
parts[c : conicalTestTube[cylindrical_, conical_, cap_]] :=
 {"cylindrical" → cylindrical, "conical" → conical, "cap" → cap} // Association
parts[c : conicalTestTube[idTop_, idHip_, idBottom_, hTop_, hBottom_]] := parts @ toCanonical @ c
test @ parts[toCanonical @ conicalTestTube[{idTop, idHip, idBottom}, {hTop, hBottom}]];
```

```
parts[toCanonical[conicalTestTube[{idTop, idHip, idBottom}, {hTop, hBottom}]]] →
 <|cylindrical → invertedFrustum[hTop, idTop/2, idHip/2],
  conical → invertedFrustum[hBottom - idBottom, idHip/2, idBottom/2], cap → invertedSphericalCap[idBottom/2, idBottom/2]|>
```

Volume

```
volume[c : conicalTestTube[cylindrical_, conical_, cap_]] := Total[volume /@ parts[c]]
volume[c : conicalTestTube[idTop_, idHip_, idBottom_, hTop_, hBottom_]] := volume @ toCanonical @ c
```

Examples

The Bio-rad specs aren’t internally consistent: there’s a conflict between the well diameters and height vs the well angle.

```
example = Module[{cone, α, rsmall, rbig, hOverall, h},
  α = toRadian[17.5] / 2;
  rsmall = 2.64 / 2;
  rbig = 5.46 / 2;
  hOverall = 14.81;
  h = 14.66; (* from a previous call to Solve *)
  conicalTestTube[cylinder[hOverall - h, rbig], invertedFrustum[h, rbig, rsmall], emptyCylinder[]]]
volume @ example
Solve[% == 200, h]
```

```
conicalTestTube[cylinder[0.15, 2.73], invertedFrustum[14.66, 2.73, 1.32], cylinder[0, 0]]
```

```
200.
```

```
{}
```

If we honor the well angle, then the well diameter at opening is too small. Maybe we can’t ignore the cap?

```
example = Module[{f},
  f = invertedFrustum[h, rbig, toRadian[17.5] / 2, apexangle];
  conicalTestTube[
   cylinder[14.81 - h, rbig],
   f,
   emptyCylinder[]]]
volume @ example == 200
rsmall[parts[example]["conical"]] == 2.64 / 2
Solve[{%%, %}, {rbig, h}]
%[[2]]
example = example /. %
rbig[parts[example]["conical"]] * 2
radius[parts[example]["cylindrical"]] * 2
```

```
conicalTestTube[cylinder[14.81 - h, rbig], invertedFrustum[h, rbig, 0.152716, apexangle], cylinder[0, 0]]
```

$0.0248078\,(h - 6.4971\,rbig)^3 + (14.81 - h)\,\pi\,rbig^2 + 6.80375\,rbig^3 == 200$

$-0.153915\,h + rbig == 1.32$

Solve: Solve was unable to solve the system with inexact coefficients. The answer was obtained by solving a corresponding exact system and numericizing the result.

```
{{rbig → -1.51406, h → -18.4132}, {rbig → 2.23957, h → 5.97455}, {rbig → 4.6737, h → 21.7893}}
```

```
{rbig → 2.23957, h → 5.97455}
```

```
conicalTestTube[cylinder[8.83545, 2.23957], invertedFrustum[5.97455, 2.23957, 0.152716, apexangle], cylinder[0, 0]]
```

```
4.47914
```

```
4.47914
```

```
(tubes = {
      (* we ignore the slight widening at the throat. and the bottom cap isn't a complete hemi-sphere,
      though we treat it as such *)
      eppendorf5$0mL → Block[{side = 56.7 - 55.4, hTop = 34.12 + 2.2},
        toCanonical @ conicalTestTube[{14.8, 13.3, 3.3}, {hTop, 55.4 - hTop}]],
      eppendorf1$5ml → Block[{wall = (*measured@1000*) 10.34 - 8.81, hTop = 20},
        toCanonical @ conicalTestTube[{9.0 (*measured*), 8.7, 3.6}, {hTop, 37.8 - hTop}]],
      falcon15ml → Module[
        (* mixure of measurements and values from spec drawing *)
        (* FWIW, Opentrons uses idTop=14.9, depth=117.5. The latter is pretty good,
        given 'a' and 'wall' defined here, so our depth calc's should be good *)
        {id14, od14, wall14, wallMeasured, wall, a, b, a14, b14, c, cMeasured, d,
         bottomOd, wallCap, htopMeasured, hBottomAndCap},
        id14 = 15.0;
        od14 = 16.3;
        wall14 = od14 - id14;
        wallMeasured = 1.27;
        wall = wallMeasured;
        wallCap = 1.75;
        a = 118.8;
        b = 17.37;
        a14 = 106.3;
        b14 = 16.6;
        c = 15.75;
        cMeasured = 15.1;
        d = 22.48;
        bottomOd = 3.18;
        htopMeasured = 84.07;
        hBottomAndCap = d - wallCap;
        (* note: as defined here, we only have 14mL capacity, not 15mL. Will affect volume calc but not depth calc. *)
        toCanonical @ conicalTestTube[{b14 - (*2 - logically needed, but better fit w/o (?!)*) wall,
           cMeasured - 2 wall, bottomOd - 2 wall}, {htopMeasured, hBottomAndCap}]
       ],
      generic → toCanonical @ conicalTestTube[{idTop, idHip, idBottom}, {hTop, hBottom}],

      (* this hacks in the slightly shallower taper at the top, which isn't sized on the spec drawing *)
      bioradPlateWell → Module[{hCyl = 0.15, rbig = 5.46 / 2, rsmall = 2.64 / 2, cyl, con, cap},
        cyl = cylinder[hCyl, rbig];
        con = invertedFrustum[14.81 - hCyl, rbig, rsmall];
        cap = emptyCylinder[];
        conicalTestTube[cyl, con, cap]],

      (* see above *)
      bioradPlateWell2 → conicalTestTube[cylinder[8.835453539401207`, 2.239570651942052`],
        invertedFrustum[5.974546460598792`, 2.239570651942052`, 0.15271630954950383`, apexangle], cylinder[0, 0]],

      idtTube → conicalTestTube[
        cylinder[40.73, 8.31 / 2],
        invertedCone[3.2, 8.31 / 2],
        emptyCylinder[]
       ]
     } // Association) // Normal // ColumnForm
test [parts[tubes[#]]] &/@ Keys[tubes];
test [volume[tubes[#]]] &/@ Keys[tubes];
```

```
eppendorf5$0mL → conicalTestTube[invertedFrustum[36.32, 7.4, 6.65], invertedFrustum[15.78, 6.65, 1.65], invertedSphericalCap[1.65, 1.
eppendorf1$5ml → conicalTestTube[invertedFrustum[20, 4.5, 4.35], invertedFrustum[14.2, 4.35, 1.8], invertedSphericalCap[1.8, 1.8]]
falcon15ml → conicalTestTube[invertedFrustum[84.07, 7.665, 6.28], invertedFrustum[20.09, 6.28, 0.32], invertedSphericalCap[0.32, 0.32
generic → conicalTestTube[invertedFrustum[hTop, idTop/2, idHip/2], invertedFrustum[hBottom - idBottom, idHip/2, idBottom/2], invertedSphericalCap[
bioradPlateWell → conicalTestTube[cylinder[0.15, 2.73], invertedFrustum[14.66, 2.73, 1.32], cylinder[0, 0]]
bioradPlateWell2 → conicalTestTube[cylinder[8.83545, 2.23957], invertedFrustum[5.97455, 2.23957, 0.152716, apexangle], cylinder[0, 0]
idtTube → conicalTestTube[cylinder[40.73, 4.155], invertedCone[3.2, 4.155], cylinder[0, 0]]
```

```
parts[tubes[eppendorf5$0mL]] → ⟨|cylindrical → invertedFrustum[36.32, 7.4, 6.65],
  conical → invertedFrustum[15.78, 6.65, 1.65], cap → invertedSphericalCap[1.65, 1.65]|⟩
```

```
parts[tubes[eppendorf1$5ml]] →
 ⟨|cylindrical → invertedFrustum[20, 4.5, 4.35], conical → invertedFrustum[14.2, 4.35, 1.8], cap → invertedSphericalCap[1.8, 1.8]|⟩
```

```
parts[tubes[falcon15ml]] → ⟨|cylindrical → invertedFrustum[84.07, 7.665, 6.28],
  conical → invertedFrustum[20.09, 6.28, 0.32], cap → invertedSphericalCap[0.32, 0.32]|⟩
```

parts[tubes[generic]] → ⟨| cylindrical → invertedFrustum$\left[\text{hTop}, \frac{\text{idTop}}{2}, \frac{\text{idHip}}{2}\right]$,
conical → invertedFrustum$\left[\text{hBottom} - \text{idBottom}, \frac{\text{idHip}}{2}, \frac{\text{idBottom}}{2}\right]$, cap → invertedSphericalCap$\left[\frac{\text{idBottom}}{2}, \frac{\text{idBottom}}{2}\right]$ |⟩

```
parts[tubes[bioradPlateWell]] →
 ⟨|cylindrical → cylinder[0.15, 2.73], conical → invertedFrustum[14.66, 2.73, 1.32], cap → cylinder[0, 0]|⟩
```

```
parts[tubes[bioradPlateWell2]] → ⟨|cylindrical → cylinder[8.83545, 2.23957],
  conical → invertedFrustum[5.97455, 2.23957, 0.152716, apexangle], cap → cylinder[0, 0]|⟩
```

```
parts[tubes[idtTube]] → ⟨|cylindrical → cylinder[40.73, 4.155], conical → invertedCone[3.2, 4.155], cap → cylinder[0, 0]|⟩
```

```
volume[tubes[eppendorf5$0mL]] → 6602.87
```

```
volume[tubes[eppendorf1$5ml]] → 1688.61
```

```
volume[tubes[falcon15ml]] → 13756.5
```

volume[tubes[generic]] →

$$\frac{\text{idBottom}^3 \pi}{12} + \frac{1}{12} (\text{hBottom} - \text{idBottom}) \left(\text{idBottom}^2 + \text{idBottom idHip} + \text{idHip}^2\right) \pi + \frac{1}{12} \text{hTop} \left(\text{idHip}^2 + \text{idHip idTop} + \text{idTop}^2\right) \pi$$

```
volume[tubes[bioradPlateWell]] → 200.
```

```
volume[tubes[bioradPlateWell2]] → 200.
```

```
volume[tubes[idtTube]] → 2266.91
```

Height & Depth

Math

```
depthFromVolume[c : conicalTestTube[{idTop_, idHip_, idBottom_}, {hTop_, hBottom_}], v_] := depthFromVolume[toCanonical @ c, v]
depthFromVolume[c : conicalTestTube[cylindrical_, conical_, cap_], v_] :=
 Module[{vCylindrical, vConical, vCap, dFromCap, dFromConical, dOther, result},
  vCap = volume[cap];
  vConical = volume[conical];
  dFromCap = depthFromVolume[cap, v];
  dFromConical = height[cap] + depthFromVolume[conical, v - vCap];
  dOther = height[cap] + height[conical] + depthFromVolume[cylindrical, v - vCap - vConical];
  Piecewise[
   {
    {dFromCap, v ≤ vCap},
    {dFromConical, v ≤ vConical},
    {dOther, True}
   }
  ]
 ]
```

Calibrating against known tubes

```
test @ depthFromVolume[tubes[eppendorf1$5ml], 500];
test @ depthFromVolume[tubes[eppendorf1$5ml], 1500];
test @ (depthFromVolume[tubes[eppendorf1$5ml], 1500] - depthFromVolume[tubes[eppendorf1$5ml], 1000]);
```

```
depthFromVolume[tubes[eppendorf1$5ml], 500] → 16.7021
```

```
depthFromVolume[tubes[eppendorf1$5ml], 1500] → 33.0204
```

```
depthFromVolume[tubes[eppendorf1$5ml], 1500] - depthFromVolume[tubes[eppendorf1$5ml], 1000] → 8.0461
```

```
test @ depthFromVolume[tubes[eppendorf5$0mL], 5000];
```

```
depthFromVolume[tubes[eppendorf5$0mL], 5000] → 44.1795
```

```
test @ tubes[falcon15ml];
test @ depthFromVolume[tubes[falcon15ml], 3000];
test @ depthFromVolume[tubes[falcon15ml], 14 000];
test @ (depthFromVolume[tubes[falcon15ml], 14 000] - depthFromVolume[tubes[falcon15ml], 2000](* measured at 76.5*));
```

```
tubes[falcon15ml] →
 conicalTestTube[invertedFrustum[84.07, 7.665, 6.28], invertedFrustum[20.09, 6.28, 0.32], invertedSphericalCap[0.32, 0.32]]
```

```
depthFromVolume[tubes[falcon15ml], 3000] → 36.8483
```

```
depthFromVolume[tubes[falcon15ml], 14 000] → 105.795
```

```
depthFromVolume[tubes[falcon15ml], 14 000] - depthFromVolume[tubes[falcon15ml], 2000] → 76.5075
```

```
test @ tubes[bioradPlateWell];
test @ depthFromVolume[tubes[bioradPlateWell], 84];
test @ depthFromVolume[tubes[bioradPlateWell], 84 - 50];
test @ toDeg @ apexangle @ parts[tubes[bioradPlateWell]]["conical"];
```

```
tubes[bioradPlateWell] → conicalTestTube[cylinder[0.15, 2.73], invertedFrustum[14.66, 2.73, 1.32], cylinder[0, 0]]
```

```
depthFromVolume[tubes[bioradPlateWell], 84] → 8.68692
```

```
depthFromVolume[tubes[bioradPlateWell], 84 - 50] → 4.54217
```

```
toDeg[apexangle[parts[tubes[bioradPlateWell]][conical]]] → 5.49381
```

```
test @ tubes[bioradPlateWell2];
test @ depthFromVolume[tubes[bioradPlateWell2], 84];
test @ depthFromVolume[tubes[bioradPlateWell2], 84 - 50];
test @ toDeg @ apexangle @ parts[tubes[bioradPlateWell2]]["conical"];
```

```
tubes[bioradPlateWell2] →
 conicalTestTube[cylinder[8.83545, 2.23957], invertedFrustum[5.97455, 2.23957, 0.152716, apexangle], cylinder[0, 0]]
```

```
depthFromVolume[tubes[bioradPlateWell2], 84] → 7.44829
```

```
depthFromVolume[tubes[bioradPlateWell2], 84 - 50] → 4.0258
```

```
toDeg[apexangle[parts[tubes[bioradPlateWell2]][conical]]] → 8.75
```

```
test @ depthFromVolume[tubes[idtTube], 250];
test @ (depthFromVolume[tubes[idtTube], 1250] - depthFromVolume[tubes[idtTube], 250]);
```

```
depthFromVolume[tubes[idtTube], 250] → 6.74277
```

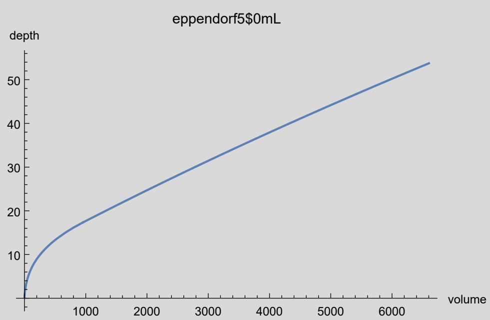
```
depthFromVolume[tubes[idtTube], 1250] - depthFromVolume[tubes[idtTube], 250] → 18.4378
```

For volume as parameter

```
printAndPlot[name_] := Module[{expr},
  If[ToString[name] == "generic",
   test @ depthFromVolume[tubes[name], vol];,
   test @ N @ depthFromVolume[tubes[name], vol];
   expr = N @ depthFromVolume[tubes[name], vol];
   printCell @ Plot[expr, {vol, 0, volume[tubes[name]]}, AxesLabel → {"volume", "depth"}, PlotLabel → name]
  ]]
printAndPlot /@ Keys[tubes];
```

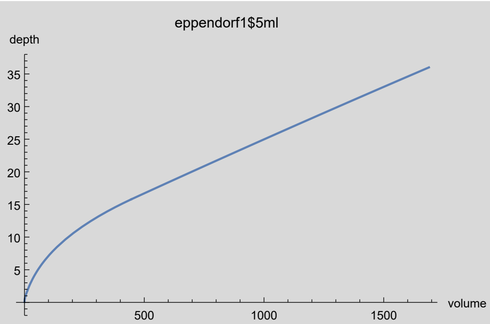
N[depthFromVolume[tubes[eppendorf5$0mL], vol]] →

$$\begin{cases} 1.65 - \dfrac{2.51187 - 4.35069\,i}{\left(28.2249 - 3.\,vol + 1.73205\sqrt{-56.4497\,vol + 3.\,vol^2}\right)^{1/3}} - (0.270963 + 0.469322\,i)\left(28.2249 - 3.\,vol + 1.73205\sqrt{-56.4497\,vol + 3.\,vol^2}\right)^{1/3} & vol \le 9.40828 \\ -3.5574 + 1.25825\,(25.9645 + 4.77465\,vol)^{1/3} & vol \le 957.074 \\ -304.607 + 14.623\,(9988.78 + 0.716197\,vol)^{1/3} & True \end{cases}$$



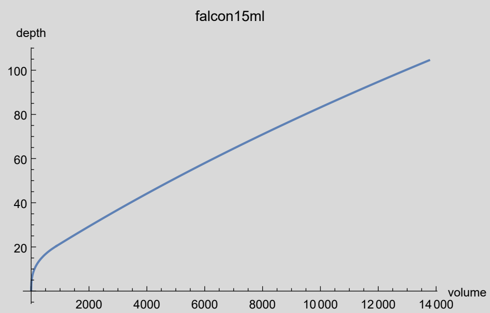
N[depthFromVolume[tubes[eppendorf1$5ml], vol]] →

$$\begin{cases} 1.8 - \dfrac{2.98934 - 5.17768\,i}{\left(36.6435 - 3.\,vol + 1.73205\sqrt{-73.2871\,vol + 3.\,vol^2}\right)^{1/3}} - (0.270963 + 0.469322\,i)\left(36.6435 - 3.\,vol + 1.73205\sqrt{-73.2871\,vol + 3.\,vol^2}\right)^{1/3} & vol \le 12.2145 \\ -8.22353 + 2.2996\,(53.0712 + 2.43507\,vol)^{1/3} & vol \le 445.995 \\ -564. + 49.1204\,(1580.62 + 0.143239\,vol)^{1/3} & True \end{cases}$$



`N[depthFromVolume[tubes[falcon15ml], vol]] →`

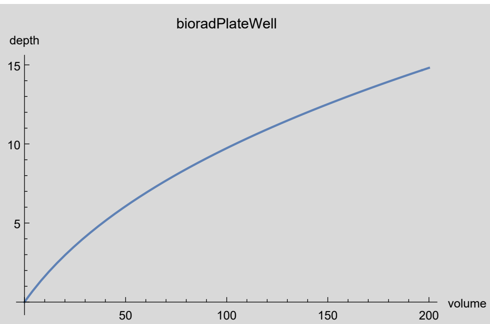
$$\begin{cases} 0.32-\dfrac{0.0944778-0.16364\,i}{\left(0.205887-3.\,\text{vol}+1.73205\sqrt{-0.411775\,\text{vol}+3.\,\text{vol}^2}\right)^{1/3}}-\left(0.270963+0.469322\,i\right)\left(0.205887-3.\,\text{vol}+1.73205\sqrt{-0.411775\,\text{vol}+3.\,\text{vol}^2}\right)^{1/3} & \text{vol}\le 0.0686291 \\ -0.758658+1.23996\,(0.267715+5.69138\,\text{vol})^{1/3} & \text{vol}\le 874.146 \\ -360.788+13.8562\,(19665.7+1.32258\,\text{vol})^{1/3} & \text{True} \end{cases}$$



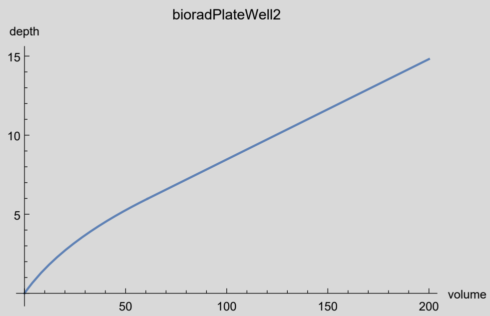
`depthFromVolume[tubes[generic], vol] →`

$$\begin{cases} \dfrac{\text{idBottom}}{2}-\dfrac{\left(1-i\sqrt{3}\right)\text{idBottom}^2\,\pi^{1/3}}{4\times 2^{2/3}\left(\frac{\text{idBottom}^3\pi}{4}-3\,\text{vol}+\sqrt{3}\sqrt{-\frac{1}{2}\text{idBottom}^3\,\pi\,\text{vol}+3\,\text{vol}^2}\right)^{1/3}}-\dfrac{\left(1+i\sqrt{3}\right)\left(\frac{\text{idBottom}^3\pi}{4}-3\,\text{vol}+\sqrt{3}\sqrt{-\frac{1}{2}\text{idBottom}^3\,\pi\,\text{vol}+3\,\text{vol}^2}\right)^{1/3}}{2\,(2\pi)^{1/3}} & \text{vol}\le\dfrac{\text{idBottom}^3\,\pi}{12} \\ \dfrac{\text{idBottom}}{2}-\dfrac{1}{\text{idBottom}-\text{idHip}}\left(-\text{hBottom idBottom}+\text{idBottom}^2+(\text{hBottom}-\text{idBottom})^{2/3}\left(\text{idBottom}^3\,(\text{hBottom}-\text{idHip})+\dfrac{12\,(-\text{idBottom}+\text{idHip})\,\text{vol}}{\pi}\right)^{1/3}\right) & \text{vol}\le\dfrac{1}{12}(\text{hBottom}-\text{idBottom})\left(\text{idBottom}^2+\text{idBottom idHip}+\text{idHip}^2\right)\pi \\ \text{hBottom}-\dfrac{\text{idBottom}}{2}+\dfrac{1}{\text{idHip}-\text{idTop}}\left(\text{hTop idHip}-\text{hTop}^{2/3}\left(\text{hBottom}\left(\text{idBottom}^2+\text{idBottom idHip}+\text{idHip}^2\right)(\text{idHip}-\text{idTop})+\text{idHip}\,(\text{idHip}\,(\text{hTop idHip}-\text{idBottom}\,(\text{idBottom}+\text{idHip}))+\text{idBottom}\,(\text{idBottom}+\text{idHip})\,\text{idTop})+\dfrac{12\,(-\text{idHip}+\text{idTop})\,\text{vol}}{\pi}\right)^{1/3}\right) & \text{True} \end{cases}$$

`N[depthFromVolume[tubes[bioradPlateWell], vol]] →`
$$\begin{cases} 0. & \text{vol}\le 0. \\ -13.7243+4.24819\,(33.7175+1.34645\,\text{vol})^{1/3} & \text{vol}\le 196.488 \\ 14.66-0.0427095\,(196.488-1.\,\text{vol}) & \text{True} \end{cases}$$

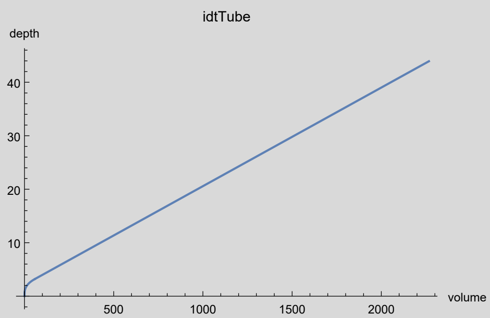


`N[depthFromVolume[tubes[bioradPlateWell2], vol]] →`
$$\begin{cases} 0. & \text{vol}\le 0. \\ -8.57618+6.4971\,(2.29997+0.146978\,\text{vol})^{1/3} & \text{vol}\le 60.7779 \\ 5.97455-0.063463\,(60.7779-1.\,\text{vol}) & \text{True} \end{cases}$$




```
N[depthFromVolume[tubes[idtTube], vol]] →
```

$$\begin{cases} 0. & vol \le 0. \\ 0.827389\, vol^{1/3} & vol \le 57.8523 \\ 3.2 - 0.0184378\, (57.8523 - 1.\, vol) & True \end{cases}$$




```
example1 = tubes[bioradPlateWell];
example2 = tubes[bioradPlateWell2];
expr1 = depthFromVolume[example1, v]
expr2 = depthFromVolume[example2, v]
Plot[{expr1, expr2}, {v, 0, volume[example1]}, AxesLabel → {"volume", "depth"}]
```

$$\begin{cases} 0 & v \le 0 \\ -13.7243 + 4.24819\, (33.7175 + 1.34645\, v)^{1/3} & v \le 196.488 \\ 14.66 - 0.0427095\, (196.488 - v) & True \end{cases}$$

$$\begin{cases} 0 & v \le 0 \\ -8.57618 + 6.4971\, (2.29997 + 0.146978\, v)^{1/3} & v \le 60.7779 \\ 5.97455 - 0.063463\, (60.7779 - v) & True \end{cases}$$