Well Geometry

Robert Atkinson
04 Oct 2019

We explore the geometry of various labware.

Basics

Cone

Inverted Cone

Cylinder

Right Conical Frustum

Inverted Right Conical Frustum

Sphere

Inverted Spherical Cap

Unknown Shape

Accessing

```
assumptions[u : unknownShape[h_, vol_]] := h ≥ 0 && vol ≥ 0
test @ assumptions[unknownShape[h, vol]];
```

```
assumptions[unknownShape[h, vol]] → h ≥ 0&&vol ≥ 0
```

```
height[u : unknownShape[h_, vol_]] := h
toCartesian[u : unknownShape[h_, vol_]] := u
```

```
volume[u : unknownShape[h_, vol_]] := Module[{},
  (*printCell[{volume, "h" → h, "vol" → vol}];*)
  vol]
```

```
depthFromVolume[u : unknownShape[h_, vol_], v_] := Module[{},
  (*printCell[{depthFromVolume, "h" → h, "vol" → vol, "v" → v}];*)
  If[v ≤ 0 || h ≤ 0 || vol ≤ 0,
   0,
   Indeterminate]]
```

Conical Test Tube

Printed by Wolfram Mathematica Student Edition

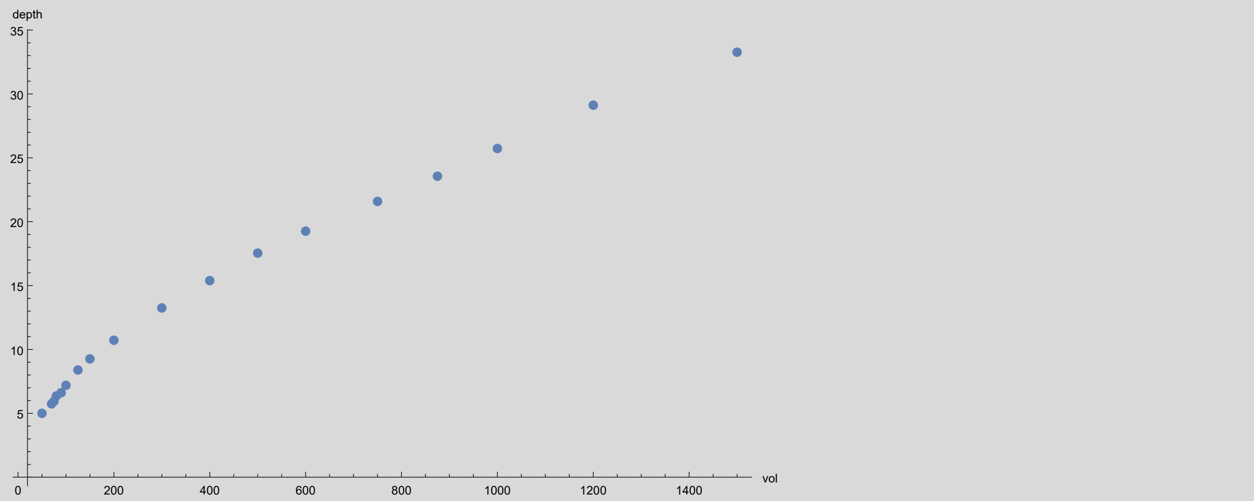
Examples

Bio-rad Deep Well Plates

Eppendorf Tubes

```
eppendorfData = ArrayReshape[{50, 5, 70, 5.74, 75, 5.94, 80, 6.36, 90, 6.61, 100, 7.19, 125, 8.39, 150, 9.26, 200, 10.72, 300,
    13.25, 400, 15.39, 500, 17.54, 600, 19.26, 750, 21.59, 875, 23.56, 1000, 25.73, 1200, 29.12, 1500, 33.27}, {18, 2}]
ListPlot[eppendorfData, ImageSize → Large, AxesLabel → {"vol", "depth"}, PlotRange → All]
```

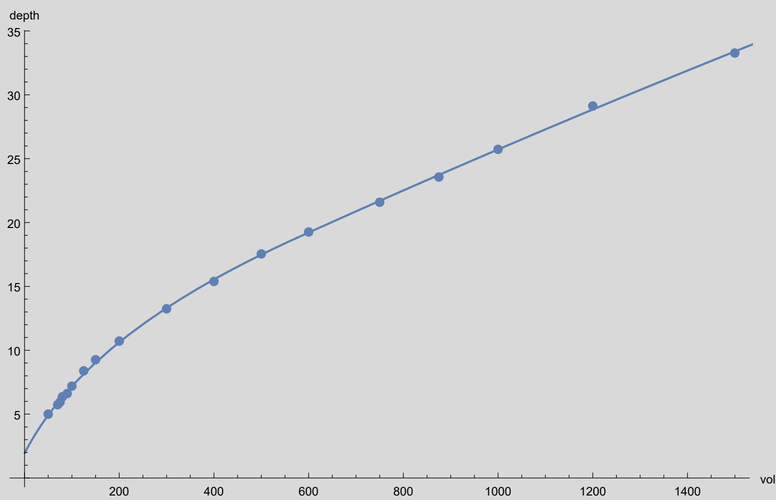
```
{{50, 5}, {70, 5.74}, {75, 5.94}, {80, 6.36}, {90, 6.61}, {100, 7.19}, {125, 8.39}, {150, 9.26}, {200, 10.72}, {300, 13.25},
 {400, 15.39}, {500, 17.54}, {600, 19.26}, {750, 21.59}, {875, 23.56}, {1000, 25.73}, {1200, 29.12}, {1500, 33.27}}
```




```
fitEppendorfData[eppendorfData_] := Module[
  {depthFunc, fit, showFit, zeroify, conicalData, conePart, coneRules, angledCone, cylinderData, offsetConicalData,
   offsetCylinderData, cylinderPart, cylinderRules, hCone, hCyl, rtop, rmid, rbottom, angledCylinder, specRules, hTot,
   tube, α, tubeRules, rconeBig, rconeSmall, wallBottom, rules, αCylinder, αCone, hCap, rCap, volCap, fittedTube},
  depthFunc[part_] := Module[{expr, v},
    expr = depthFromVolume[part, v];
    depthFunc[part] = Function[{vol}, expr /. {v → vol}]];
  fit[part_, assump_, vars_, data_] := Module[{errors, err, min, fitRules, asses},
    errors = Function[{vol, depth},
         (depthFunc[part][vol] - depth)^2
       ] @@ # & /@ data;
    err = Total[errors] // N;
    asses = assumptions[part] && (And @@ assump);
    (*test @ asses;*)
    {min, fitRules} = NMinimize[{err, asses }, vars];
    fitRules];
  showFit[part_, data_] := Module[{v},
    Show[ListPlot[{data}, ImageSize → Large, AxesLabel → {"vol", "depth"}, PlotRange → All, AxesOrigin → {0, 0}],
     Plot[depthFromVolume[part, v], {v, 0, volume[part]}]]];
  zeroify[data_] := Module[{xMin, yMin},
    {xMin, yMin} = Map[Min, Transpose @ data, {1}];
    Transpose[Transpose[data] - {xMin, yMin}]];

  conicalData = Select[eppendorfData, #[[1]] ≤ 500 &];
  cylinderData = Select[eppendorfData, #[[1]] >= 500 &]; (* hard to tell for in between data, so we're conservative *)
  offsetConicalData = zeroify[conicalData];
  offsetCylinderData = zeroify[cylinderData];
  (*printCell @ ListPlot[{conicalData, cylinderData}, ImageSize→Large, AxesLabel→{"vol", "depth"}, PlotRange→All];*)
  (*printCell @ ListPlot[{offsetCylinderData}, ImageSize→Large, AxesLabel→{"vol", "depth"}, PlotRange→All];*)
```

```
  specRules = { hTot → 37.8, rmid → 8.7 / 2, wallBottom → 38.9 - 37.8};
  printCell[specificationSays[specRules]];

  (* fit the cylinder. this gives us the apex angle of the cylinder. we don't yet know its actual height *)
  (* we dont' know rmid because the bottom of cylinderData might not be right at the mid location *)
  cylinderPart = invertedFrustum[hCyl, rtop, rmid](* /. coneRules*);
  cylinderRules = fit[cylinderPart, {hCyl > 12}, {hCyl, rtop, rmid}, offsetCylinderData];
  angledCylinder = toApexAngled[cylinderPart /. cylinderRules];
  (*test @ cylinderRules;
  test @ (cylinderPart /. cylinderRules);
  test @ angledCylinder;
  test @ toDeg @ apexangle[angledCylinder];*)
  (*printCell @ showFit[cylinderPart /. cylinderRules, offsetCylinderData];*)

  (* fit the cone. this gives us the apex angle of the cone *)
  conePart = invertedFrustum[hCone, rconeBig, rconeSmall];
  coneRules = fit[conePart, {hCone > 10}, {hCone, rconeBig, rconeSmall}, offsetConicalData];
  angledCone = toApexAngled[conePart /. coneRules];
  (*test @ coneRules;
  test @ (conePart /. coneRules);
  test @ angledCone;
  test @ toDeg @ apexangle[angledCone];*)
  (*printCell @ showFit[conePart /. coneRules, offsetConicalData];*)

  (* summarize what we know *)
  rules = {αCylinder → apexangle[angledCylinder], αCone → apexangle[angledCone]};
  (*test @ rules;*)

  (* put these together. *)
  (* Cap is just a shape that can fix a volume; we have no data in that range, and can't measure volumes therein. *)
  tube = conicalTestTube[
    (invertedFrustum[hCyl, rbig[hCyl, rmid, αCylinder, apexangle], αCylinder, apexangle] /. rules),
    (invertedFrustum[hCone, rmid, αCone, apexangle] /. rules),
    (unknownShape[hCap, volCap])
   ];
  tube = tube /. { hCone → (hTot /. specRules) - hCyl - hCap};
  (*test @ tube;*)
  tubeRules =
   fit[tube, {hCap < 5, hCyl > 10, rmid > 4, rmid < 6(*, rCap ≥ hCap*)}, {hCyl, rmid, hCap, volCap}, eppendorfData];
  fittedTube = toCartesian[tube /. tubeRules];
  (*test @ tubeRules;
  test @ fittedTube;*)
  printCell @ showFit[fittedTube, eppendorfData];
  fittedTube
 ]
fittedEppendorf = fitEppendorfData[eppendorfData]
test @ height @ fittedEppendorf;
test @ depthFromVolume[fittedEppendorf, volume[fittedEppendorf]];
test @ volume @ fittedEppendorf;
```

```
specificationSays[{hTot$116829 → 37.8, rmid$116829 → 4.35, wallBottom$116829 → 1.1}]
```




```
conicalTestTube[invertedFrustum[18.9894, 4.70751, 4.35636],
 invertedFrustum[16.8419, 4.35636, 2.1099], unknownShape[1.96866, 0.550217]]
```

```
height[fittedEppendorf] → 37.8
```

```
depthFromVolume[fittedEppendorf, volume[fittedEppendorf]] → 37.8
```

```
volume[fittedEppendorf] → 1801.76
```

It’s regrettable that we don’t bottom out at 0 mm (we bottom out at about 2 mm), but the data does really fit quite nicely otherwise.

It should be noted that the specification indicates that the upper ‘cylindrical’ inverted frustum isn’t actually an inverted frustum but has a bit of a flare at the top.

IDT tubes

Falcon

Known Tubes

With that, we define the tubes

```
(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}]],
      fittedEppendorf1$5ml → fittedEppendorf,

      fittedFalcon15ml → fittedFalcon,
      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[]
       ],
      fittedIdtTube → fittedIdt
     } // Association) // Normal // ColumnForm
```

```
eppendorf5$0mL → conicalTestTube[invertedFrustum[36.32, 7.4, 6.65], invertedFrustum[15.78, 6.65, 1.65], invertedSphericalCap[1.65, 1.6
eppendorf1$5ml → conicalTestTube[invertedFrustum[20, 4.5, 4.35], invertedFrustum[14.2, 4.35, 1.8], invertedSphericalCap[1.8, 1.8]]
fittedEppendorf1$5ml → conicalTestTube[invertedFrustum[18.9894, 4.70751, 4.35636], invertedFrustum[16.8419, 4.35636, 2.1099], unknown
fittedFalcon15ml → conicalTestTube[invertedFrustum[95.9755, 7.42952, 6.65602], invertedFrustum[22.0945, 6.65602, 1.14806], cylinder[0
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]]
fittedIdtTube → conicalTestTube[cylinder[38.3037, 4.16389], invertedCone[3.69629, 4.16389], cylinder[0, 0]]
```

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[fittedEppendorf1$5ml], 500];
test @ depthFromVolume[tubes[fittedEppendorf1$5ml], 1500];
test @ (depthFromVolume[tubes[fittedEppendorf1$5ml], 1500] - depthFromVolume[tubes[eppendorf1$5ml], 1000]);
```

```
depthFromVolume[tubes[fittedEppendorf1$5ml], 500] → 17.4848
```

```
depthFromVolume[tubes[fittedEppendorf1$5ml], 1500] → 33.3897
```

```
depthFromVolume[tubes[fittedEppendorf1$5ml], 1500] - depthFromVolume[tubes[eppendorf1$5ml], 1000] → 8.41539
```

```
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[fittedFalcon15ml];
test @ depthFromVolume[tubes[fittedFalcon15ml], 3000];
test @ depthFromVolume[tubes[fittedFalcon15ml], 14000];
test @
  (depthFromVolume[tubes[fittedFalcon15ml], 14000] - depthFromVolume[tubes[fittedFalcon15ml], 2000](* measured at 76.5*));
```

```
tubes[fittedFalcon15ml] →
 conicalTestTube[invertedFrustum[95.9755, 7.42952, 6.65602], invertedFrustum[22.0945, 6.65602, 1.14806], cylinder[0, 0]]
```

```
depthFromVolume[tubes[fittedFalcon15ml], 3000] → 34.6045
```

```
depthFromVolume[tubes[fittedFalcon15ml], 14000] → 105.188
```

```
depthFromVolume[tubes[fittedFalcon15ml], 14000] - depthFromVolume[tubes[fittedFalcon15ml], 2000] → 77.6146
```

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

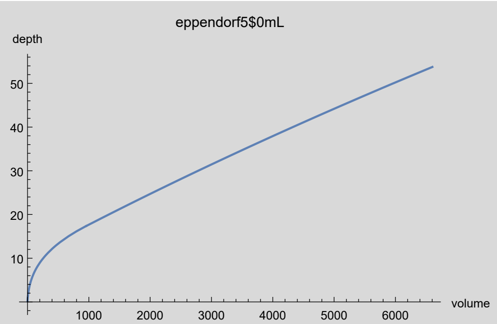
eppendorf5$0mL

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

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

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

N[depthFromVolume[tubes[eppendorf5$0mL], volume[tubes[eppendorf5$0mL]]]] → 53.75



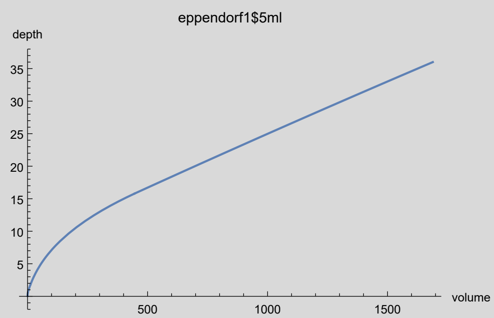
eppendorf1$5ml

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

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

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

N[depthFromVolume[tubes[eppendorf1$5ml], volume[tubes[eppendorf1$5ml]]]] → 36.

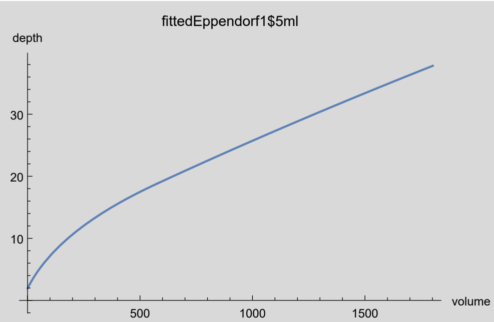


fittedEppendorf1$5ml

```
N[depthFromVolume[tubes[fittedEppendorf1$5ml], vol]] → {
  If[vol ≤ 0., 0., Indeterminate]                          vol ≤ 0.550217
  -13.8495 + 2.9248 (157.009 + 2.14521 vol)^(1/3)          vol ≤ 575.33
  -216.767 + 20.2694 (1376.83 + 0.33533 vol)^(1/3)         True
```

```
N[volume[tubes[fittedEppendorf1$5ml]]] → 1801.76
```

```
N[depthFromVolume[tubes[fittedEppendorf1$5ml], volume[tubes[fittedEppendorf1$5ml]]]] → 37.8
```

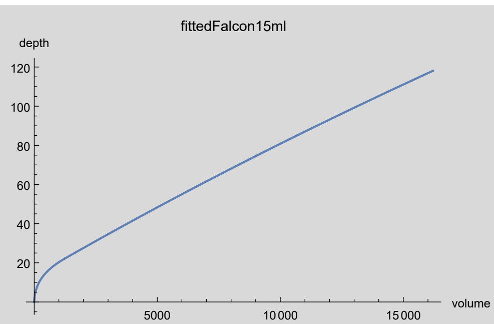


fittedFalcon15ml

```
N[depthFromVolume[tubes[fittedFalcon15ml], vol]] → {
  0.                                                        vol ≤ 0.
  -4.60531 + 1.42955 (33.4335 + 5.25971 vol)^(1/3)         vol ≤ 1232.34
  -803.774 + 27.1004 (27390.9 + 0.738644 vol)^(1/3)        True
```

```
N[volume[tubes[fittedFalcon15ml]]] → 16202.8
```

```
N[depthFromVolume[tubes[fittedFalcon15ml], volume[tubes[fittedFalcon15ml]]]] → 118.07
```



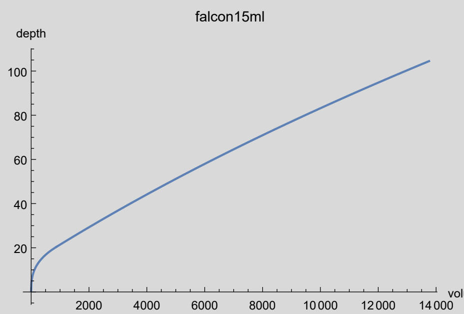
falcon15ml

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

$$\begin{cases} 0.32 - \dfrac{0.0944778 - 0.16364\,i}{\left(0.205887 - 3.\,vol + 1.73205\sqrt{-0.411775\,vol + 3.\,vol^2}\right)^{1/3}} - (0.270963 + 0.469322\,i)\left(0.205887 - 3.\,vol + 1.73205\sqrt{-0.411775\,vol + 3.\,vol^2}\right)^{1/3} & vol \le 0.0686291 \\ -0.758658 + 1.23996\,(0.267715 + 5.69138\,vol)^{1/3} & vol \le 874.146 \\ -360.788 + 13.8562\,(19\,665.7 + 1.32258\,vol)^{1/3} & True \end{cases}$$

```
N[volume[tubes[falcon15ml]]] → 13 756.5
```

```
N[depthFromVolume[tubes[falcon15ml], volume[tubes[falcon15ml]]]] → 104.48
```



generic

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

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

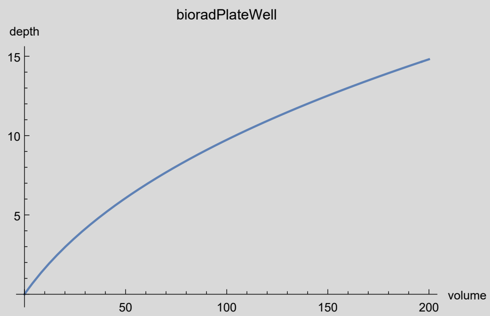
bioradPlateWell

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

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

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

```
N[depthFromVolume[tubes[bioradPlateWell], volume[tubes[bioradPlateWell]]]] → 14.81
```

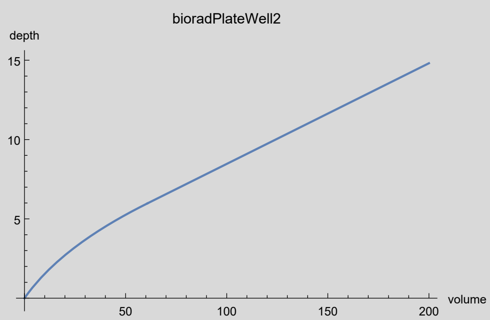


bioradPlateWell2

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

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

N[depthFromVolume[tubes[bioradPlateWell2], volume[tubes[bioradPlateWell2]]]] → 14.81

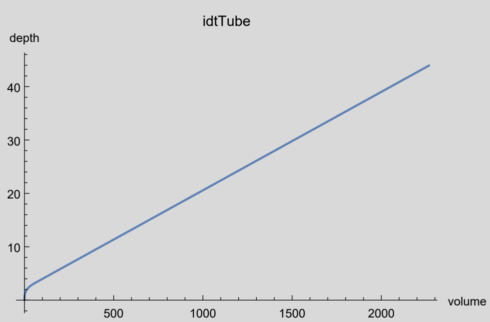


idtTube

N[depthFromVolume[tubes[idtTube], vol]] → $\begin{cases} 0. & vol \le 0. \\ 0.827389\,vol^{1/3} & vol \le 57.8523 \\ 3.2 - 0.0184378\,(57.8523 - 1.\,vol) & True \end{cases}$

N[volume[tubes[idtTube]]] → 2266.91

N[depthFromVolume[tubes[idtTube], volume[tubes[idtTube]]]] → 43.93

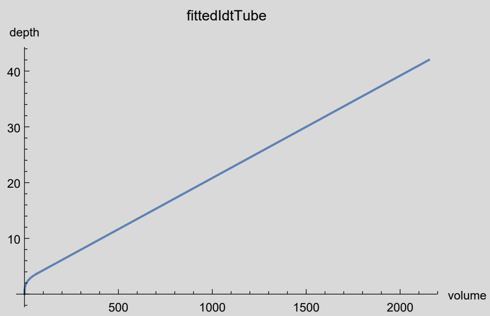


fittedIdtTube

N[depthFromVolume[tubes[fittedIdtTube], vol]] → $\begin{cases} 0. & vol \le 0. \\ 0.909568\,vol^{1/3} & vol \le 67.1109 \\ 3.69629 - 0.0183591\,(67.1109 - 1.\,vol) & True \end{cases}$

```
N[volume[tubes[fittedIdtTube]]] → 2153.47
```

```
N[depthFromVolume[tubes[fittedIdtTube], volume[tubes[fittedIdtTube]]]] → 42.
```



Comparing 1.5 mL Eppendorf Tube Models

The fitted Eppendorf model clearly is better.

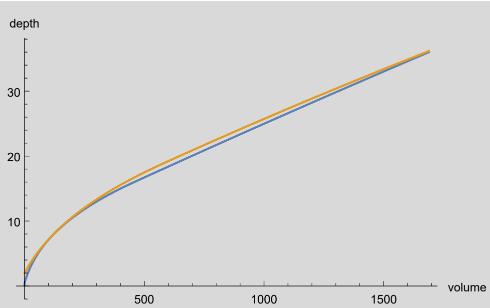
```
example1 = tubes[eppendorf1$5ml];
example2 = tubes[fittedEppendorf1$5ml];
test @ example1;
test @ example2;
expr1 = depthFromVolume[example1, v]
expr2 = depthFromVolume[example2, v]
Plot[{expr1, expr2}, {v, 0, volume[example1]}, AxesLabel → {"volume", "depth"}]
Plot[expr1 - expr2, {v, 0, volume[example1]}, AxesLabel → {"volume", "Δdepth"}]
Show[ListPlot[{eppendorfData}, AxesLabel → {"vol", "depth"}, PlotRange → All, AxesOrigin → {0, 0}, ImageSize → Large],
 Plot[{depthFromVolume[example1, v], depthFromVolume[example2, v]}, {v, 0, volume[example1]}]]
```

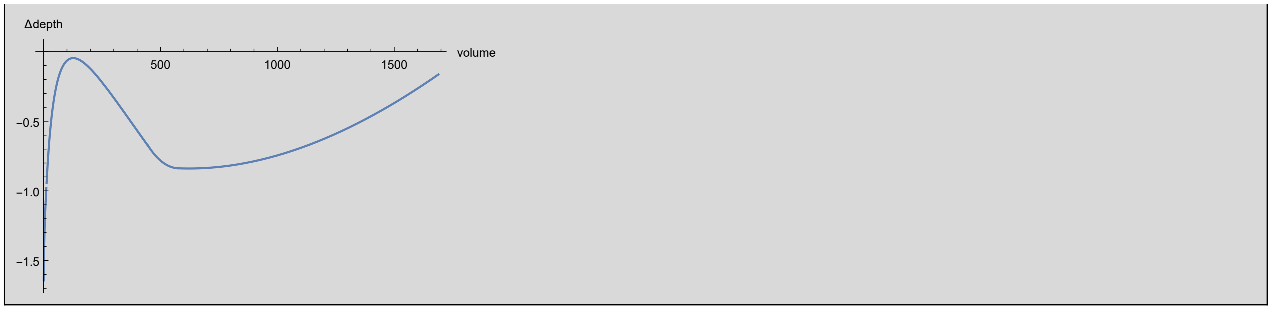
```
example1 → conicalTestTube[invertedFrustum[20, 4.5, 4.35], invertedFrustum[14.2, 4.35, 1.8], invertedSphericalCap[1.8, 1.8]]
```

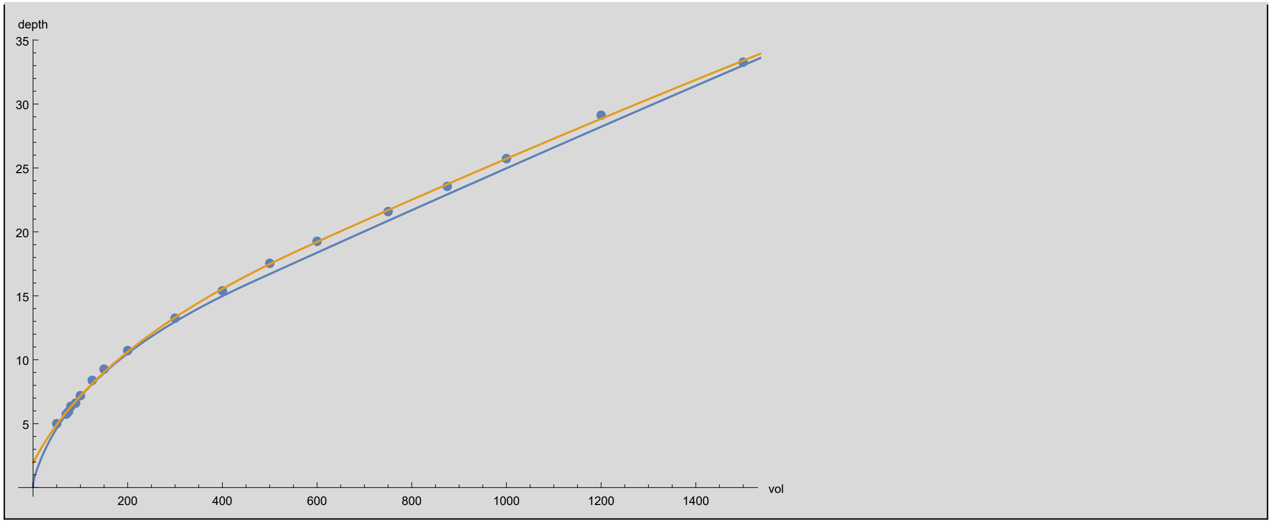
```
example2 → conicalTestTube[invertedFrustum[18.9894, 4.70751, 4.35636],
  invertedFrustum[16.8419, 4.35636, 2.1099], unknownShape[1.96866, 0.550217]]
```

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

$$\begin{cases} \text{If}[v \le 0, 0, \text{Indeterminate}] & v \le 0.550217 \\ -13.8495 + 2.9248\,(157.009 + 2.14521 v)^{1/3} & v \le 575.33 \\ -216.767 + 20.2694\,(1376.83 + 0.33533 v)^{1/3} & \text{True} \end{cases}$$







Comparing IDT Tube Models

The fitted IDT tube model is marginally better, but still better.

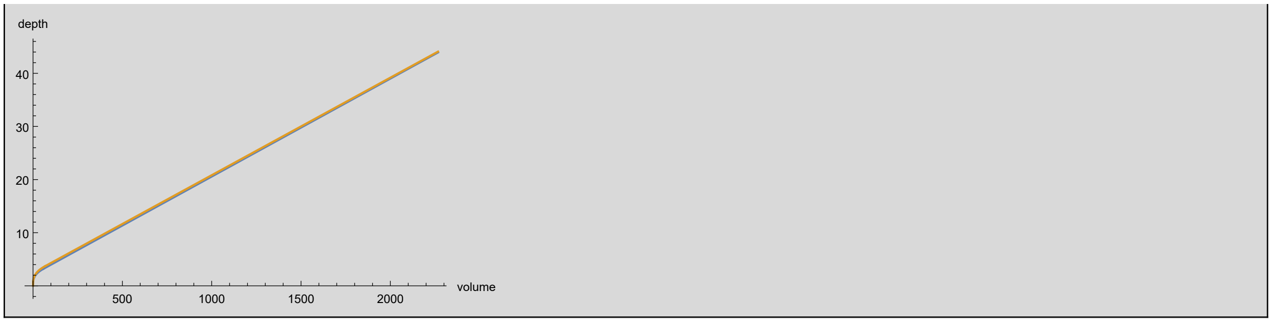
```
example1 = tubes[idtTube];
example2 = tubes[fittedIdtTube];
test @ example1;
test @ example2;
expr1 = depthFromVolume[example1, v]
expr2 = depthFromVolume[example2, v]
Plot[{expr1, expr2}, {v, 0, volume[example1]}, AxesLabel → {"volume", "depth"}]
Plot[expr1 - expr2, {v, 0, volume[example1]}, AxesLabel → {"volume", "Δdepth"}]
Show[ListPlot[{idtData}, AxesLabel → {"vol", "depth"}, PlotRange → All, AxesOrigin → {0, 0}, ImageSize → Large],
 Plot[{depthFromVolume[example1, v], depthFromVolume[example2, v]}, {v, 0, volume[example1]}]]
```

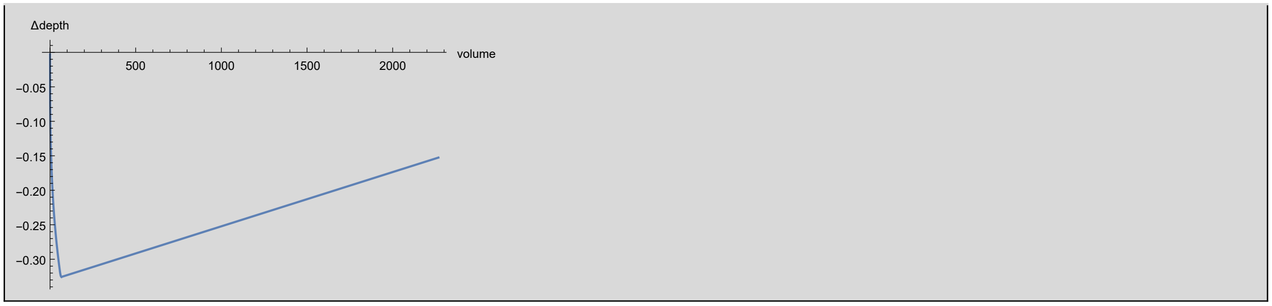
```
example1 → conicalTestTube[cylinder[40.73, 4.155], invertedCone[3.2, 4.155], cylinder[0, 0]]
```

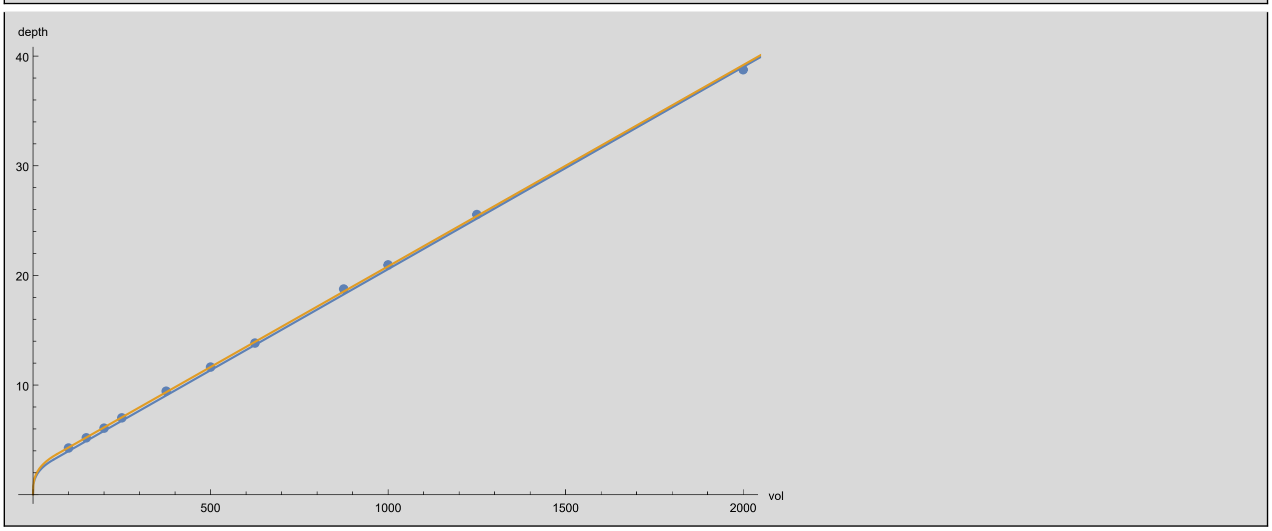
```
example2 → conicalTestTube[cylinder[38.3037, 4.16389], invertedCone[3.69629, 4.16389], cylinder[0, 0]]
```

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

$$\begin{cases} 0 & v \le 0 \\ 0.909568\, v^{1/3} & v \le 67.1109 \\ 3.69629 - 0.0183591\,(67.1109 - v) & \text{True} \end{cases}$$







Comparing Bio-rad Plate models

Which should we use? At the moment it’s unclear.

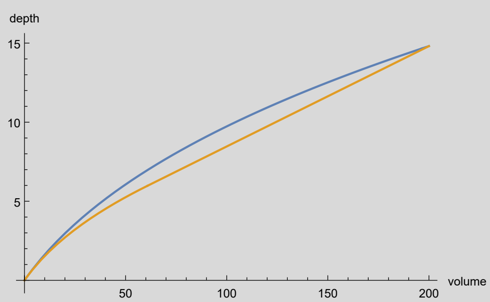
```
example1 = tubes[bioradPlateWell];
example2 = tubes[bioradPlateWell2];
test @ example1;
test @ example2;
expr1 = depthFromVolume[example1, v]
expr2 = depthFromVolume[example2, v]
Plot[{expr1, expr2}, {v, 0, volume[example1]}, AxesLabel → {"volume", "depth"}]
Plot[expr1 - expr2, {v, 0, volume[example1]},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

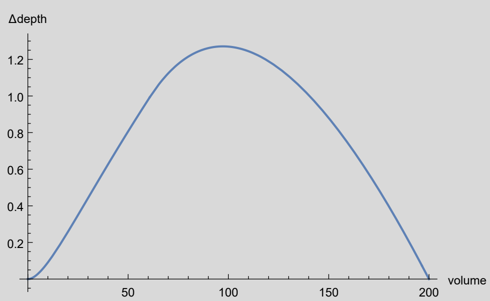
```
example1 → conicalTestTube[cylinder[0.15, 2.73], invertedFrustum[14.66, 2.73, 1.32], cylinder[0, 0]]
```

```
example2 → conicalTestTube[cylinder[8.83545, 2.23957], invertedFrustum[5.97455, 2.23957, 0.152716, apexangle], cylinder[0, 0]]
```

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

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





Comparing 15mL Falcon Tube models

We should use the fitted one, as we experimentally observed the other model predicting depths that were too large.

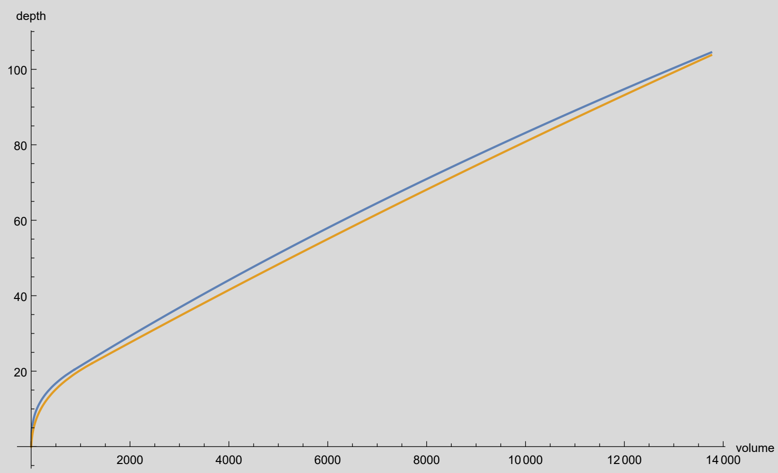
```
example1 = tubes[falcon15ml];
example2 = tubes[fittedFalcon15ml];
test @ example1;
test @ example2;
expr1 = depthFromVolume[example1, v]
expr2 = depthFromVolume[example2, v]
Plot[{expr1, expr2}, {v, 0, volume[example1]}, AxesLabel → {"volume", "depth"}, ImageSize → Large]
Plot[expr1 - expr2, {v, 0, volume[example1]}, AxesLabel → {"volume", "Δdepth"}, ImageSize → Large]
Show[ListPlot[{falconData}, AxesLabel → {"vol", "depth"}, PlotRange → All, AxesOrigin → {0, 0}, ImageSize → Large],
 Plot[{depthFromVolume[example1, v], depthFromVolume[example2, v]}, {v, 0, volume[example1]}]]
```

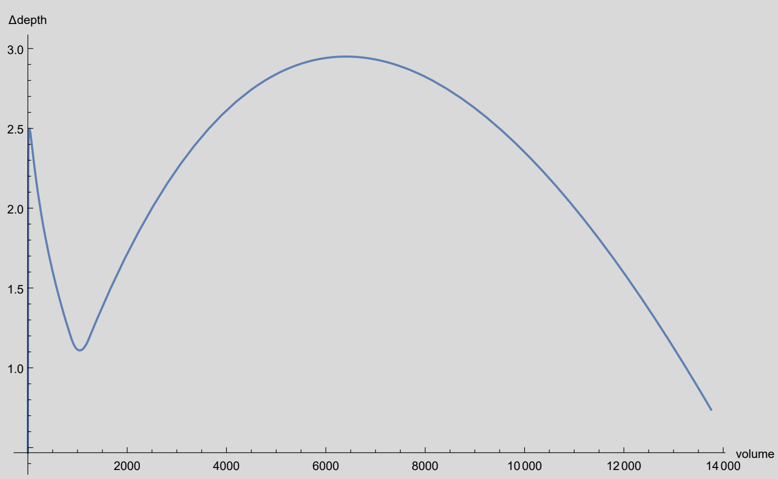
```
example1 →
 conicalTestTube[invertedFrustum[84.07, 7.665, 6.28], invertedFrustum[20.09, 6.28, 0.32], invertedSphericalCap[0.32, 0.32]]
```

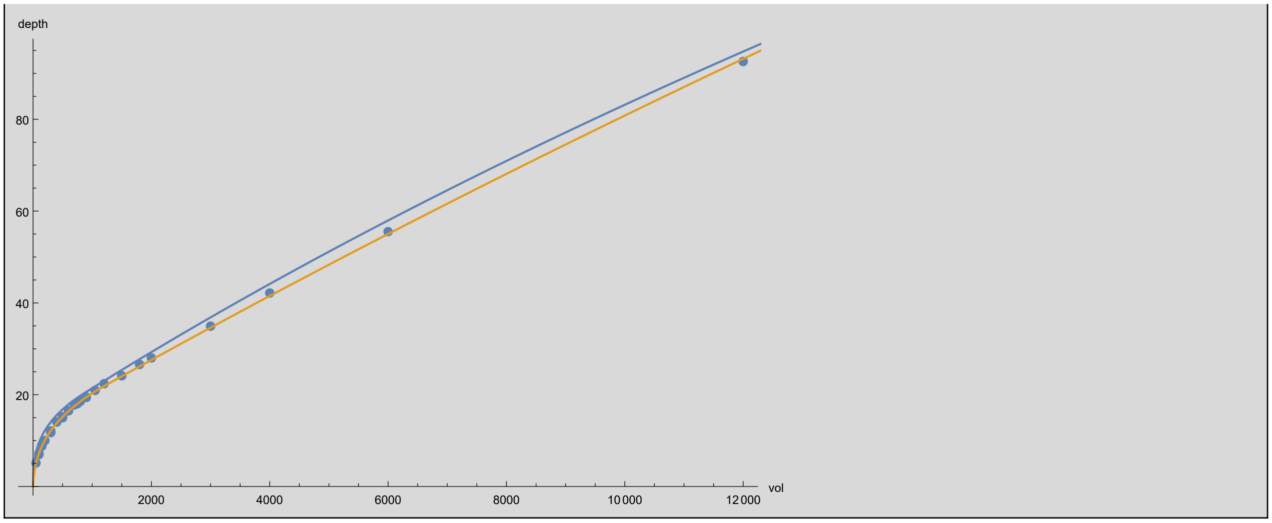
```
example2 → conicalTestTube[invertedFrustum[95.9755, 7.42952, 6.65602], invertedFrustum[22.0945, 6.65602, 1.14806], cylinder[0, 0]]
```

$$\begin{cases} 0.32 - \frac{0.0944778 - 0.16364\, i}{\left(0.205887 - 3\, v + \sqrt{3}\, \sqrt{-0.411775\, v + 3\, v^2}\right)^{1/3}} - \frac{\left(1 + i\sqrt{3}\right)\left(0.205887 - 3\, v + \sqrt{3}\, \sqrt{-0.411775\, v + 3\, v^2}\right)^{1/3}}{2\,(2\,\pi)^{1/3}} & v \le 0.0686291 \\ -0.758658 + 1.23996\,(0.267715 + 5.69138\, v)^{1/3} & v \le 874.146 \\ -360.788 + 13.8562\,(19\,665.7 + 1.32258\, v)^{1/3} & \text{True} \end{cases}$$

$$\begin{cases} 0 & v \le 0 \\ -4.60531 + 1.42955\,(33.4335 + 5.25971\, v)^{1/3} & v \le 1232.34 \\ -803.774 + 27.1004\,(27\,390.9 + 0.738644\, v)^{1/3} & \text{True} \end{cases}$$






depth
80
60
40
20
2000
4000
6000
8000
10 000
12 000
vol