V-Berth Volume

The Jupyter Book community

May 27, 2021

CONTENTS

This is an example of a fairly complex computation that benefits from a spreadsheet-like capability.

We want to change the measurements, and recompute a very complex estimation.

This use case of “enter data – see resulting computation” works nicely for spreadsheets where the computations aren’t too tricky. We can examine them for correctness, and trust they are working.

When the computations become non-trivial, a spreadsheet can be challenging to validate and debug. We may not be able to simply examine a computation cell to be sure it’s right.

This is where Jupyter Lab can be a dramatic improvement over a spreadsheet. In these examples we’ll do complex symbolic math (“algebra”) as well as ordinary applied math to compute useful values.

CHAPTER

ONE

COMPONENTS REQUIRED

You'll need Python, of course. You may want to use Miniconda to help install Python.

```
https://docs.conda.io/en/latest/miniconda.html
```

With miniconda, you can build your Python environment.

```
conda create --name model python=3.9
conda activate model
```

Once you have Python, you can then install the suite of tools for working with more advanced symbolic math, Sympy https://docs.sympy.org/latest/index.html

```
python -m pip install sympy jupyter-lab
```

With the Jupyter lab you can launch the `.ipynb` notebooks to make changes, and do computations. Like a spreadsheet. But with real code and real math.

The final publication of a report or book can be done with two additional tools.

Jupyter {Book} https://jupyterbook.org/intro.html

```
python -m pip install jupyter-book
```

Requires MyST{NB} https://myst-nb.readthedocs.io/en/latest/

```
python -m pip install myst-nb
```

CHAPTER

TWO

DOCUMENT PRODUCTION

Run the following:

```
jupyter-book build .
cp _build/html docs
```

This will create the HTML version.

To make a PDF, use the following.

```
jupyter-book build . --builder pdflatex
```

(This generally relies on having TeX installed.)

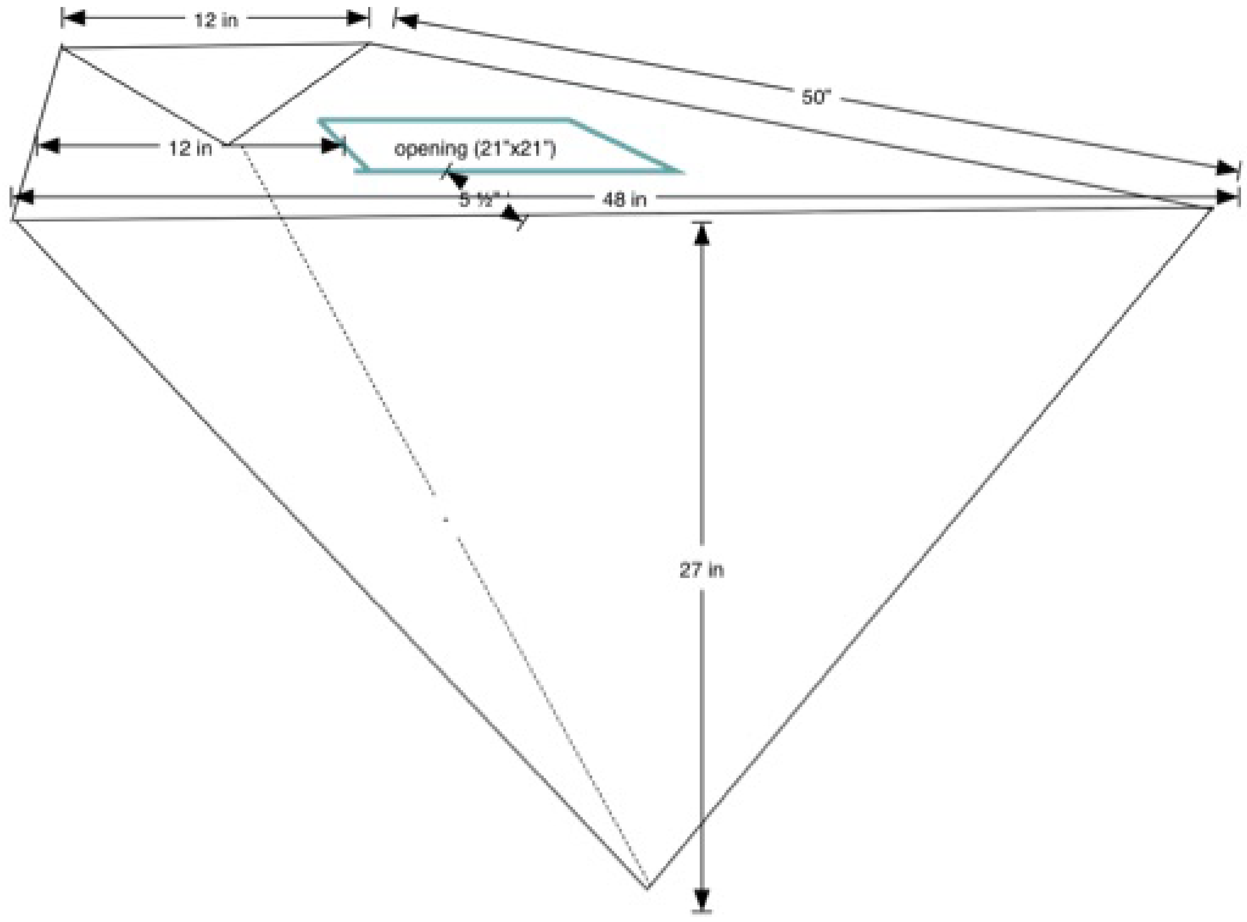
Fair warning: the structure used in this first release doesn't look very good in a PDF. An outline that's accetable in HTML can turn out to be weirdly structured when viewed as a PDF.

2.1 Volume of the V-Berth Tank

The objective is to compute the volume of a water tank under the V-berth on a boat.

For the non-nautical, the V-berth is the v-shaped bedroom built into the bow of the boat. This is a complex shape, flat on the top, but V-shaped on the bottom. It sweeps up at the forward end, almost coming to a point.

This isn't easy to visualize. The following sketch views the tank from the aft end looking forward, toward the pointy end of the boat. It shows how there's a triangular face aft, and a much smaller triangular face forward. The top tapers from the aft end, and is a kind of truncated triangle. The sides slope down along the inside of the boat's hull.



This diagram is facing forward, showing the large triangle at the aft end of the tank. The sizes were preliminary measurements, later refined as the top was cut away to gain access to the interior.

We can describe the space in a number of ways, leading us to three models for the volume:

- **Regular Triangular Prism**. While the space tapers from the large aft end to almost a point at the forward end, we can use the midpoints along this axis to define a triangular prism that should be equivalent to the irregular shape.
- **Regular Tetrahedron**. While the tank isn't really regular, we can take the mean lengths of the six edges, and use this to describe a regular tetrahedron that should have a similar volume.
- **Irregular Triangular Prism**. The tank is a prism that tapers from aft to forward. We can describe this taper as a function of the distance along the fore-and-aft axis of the tank.

The differences, it turns out, are relatively minor. The simpler math of a triangular prism is accurate enough for our purposes. The other two models serve as confirmation of the volume.

While a single estimate of the volume is necessary, it's not sufficient. As the top of the structure was removed, it became easier to get more accurate measuremewnts. What would help is a closed form function that transforms a given collection measurements into a volume estimate.

It helps to have the computation in the form of a Jupyter Notebook. We can change the measurements and see resulting change in the volume. This lets us answer a number of design tradeoff questions regarding ways to assemble new water tanks or flexible bladders to fill the leaky old aluminum tank that's in there.

We'll start with the Triangular Prism, because it's relatively simple.

2.2 Triangular Prism

A prism is a solid with two triangular faces joined by three rectangular faces. Referring back to the diagram, this is clearly not a good description of the space. The aft and and forward end are triangular, but the remaining three faces are not rectangles; the remaining faces are irregular quadrilaterals.

The volume of a triangular prism is the area of the triangle, $\frac{1}{2}h \times w$, times the overall length of the prism, l.

$$V = \frac{\frac{h \times w}{2} \times l}{231 \textbf{ cu in/gal}}$$

Our water tank has a large triangle aft and a small triangle forward. If we take the mean between these two extreme sizes, this should describe the space.

```
from myst_nb import glue
```

```
from sympy import *
h, w, l, V = symbols('h w l V')
glue("Vol1", Eq(V, (h * w) / 2 * l / 231, evaluate=False))
```

$$V = \frac{hlw}{462}$$

Here's a slightly simplified volume of a triangular prism, V, including the conversion from cubic inches to gallons.

$$V = \frac{hlw}{462} \tag{2.1}$$

Given a width, w, height, h, and the overall length, l, we know the volume of the space, V.

Since the triangles are isoscolese, we've measured their base edge and height. We often call the base edge the "width" of the triangle in the computations that follow. If necessary, we can compute the other two edges, $e = \sqrt{h^2 + \frac{w}{2}^2}$.

2.2.1 Middling the triangles

For this model of the space, we'll take the middle of the widths and the middle of the heights to describe a regular triangular prism.

The height at the midpoint, h_m, and the width at the midpoint, w_m, are as follows:

```
var("h_f, h_a, w_f, w_a, l_fa")
```

```
(h_f, h_a, w_f, w_a, l_fa)
```

```
h_m = Rational(1, 2)*(h_f + h_a)
w_m = Rational(1, 2)*(w_f + w_a)
```

It can help to see these expressions type set and simplified. Here's what the midpoints look like.

```
h_m
```

$$\frac{h_a}{2} + \frac{h_f}{2}$$

```
w_m
```

$$\frac{w_a}{2} + \frac{w_f}{2}$$

2.2.2 Volume

The mid-point volume, V_m, is computed from the midpoint height and width, h_m and w_m, and the overall length, l_{fa}. We can expand the midpoint values with the fore-and-aft sizes to get an overall computation given the five measurements we have.

```
V_m = ((h_m * w_m)/2 * l_fa) / 231
ratsimp(V_m)
```

$$\frac{h_a l_{fa} w_a}{1848} + \frac{h_a l_{fa} w_f}{1848} + \frac{h_f l_{fa} w_a}{1848} + \frac{h_f l_{fa} w_f}{1848}$$

This form, while kind of confusing-looking, has three $h \times w \times l$ terms, giving us confidence that this will be a proper volume computation in cubic inches.

2.2.3 Measurements

The actual dimensions are as follows. (And, yes, these differ slightly from sketches shown earlier, these are actual sizes.) These are subject to change, so we've collected them in one plce.

```
measured = {
    # Forward triangle, in inches
    "h_f": 8,
    "w_f": 10 + Rational(1, 2),

    # Aft triangle, in inches
    "h_a": 27,
    "w_a": 48,

    # Overall length from forward to aft, in inches.
    "l_fa": 46,
}
```

We can substitute the measured values into the volume formula to compute the volume of the space. We'll need to reformulate this slightly, since it's a complex-looking fraction.

```
V_m.subs(measured)
```

$$\frac{4485}{88}$$

```
f"{floor(V_m.subs(measured))} {frac(V_m.subs(measured))} gallons"
```

```
'50 85/88 gallons'
```

We can define a formal equation for volume given the five measurements. This is something we can use to recompute volume as the measurements evolve.

```
glue("Vol_m", V_m.evalf(3))
```

$$0.00216 l_{fa} \left(0.5 h_a + 0.5 h_f\right) \left(0.5 w_a + 0.5 w_f\right)$$

Here's a the volume of a triangular prism, including the conversion from cubic inches to gallons.

$$0.00216 l_{fa} \left(0.5 h_a + 0.5 h_f\right) \left(0.5 w_a + 0.5 w_f\right) \tag{2.2}$$

Given the the following:

- width, w_f, and height, h_f, of the forward triangle,
- width, w_a, and height, h_a, of the aft triangle,
- and the and the overall length, l.

Finally, here's the number of gallons as a decimal number.

```
V_m.evalf(4, measured)
```

$$50.97$$

We'll use this as a baseline to compare to the other estimates. Next, we'll treat the volume as a regular tetrahedron, also. This will provide a different estimate of the volume.

2.3 Regular Tetrahedron

A tetrahedron is a structure with four triangular faces.

Because the tank is flattened at the forward end, we could define a full-sized tetrahedron, and then truncate a tiny tetrahedron from the front to better estimate the remaining volume.

A symmetric (or regular) Tetrahedron has a volume, V, based on the size of the edges, a.

$$V = \frac{a^3}{6\sqrt{2}}$$

In our case, the actual tank in question has a number of different edge lengths:

- There are three edges on the aft triangular face. We know the width and height, and can compute the other two edges.
- Of the three edges on the top triangular face, it shares an edge with the aft face. We know the width and height here, too, and can compute the remaining edges.
- Finally, the seam along the keel is the sixth and final edge. We can compute this knowing the the top and aft form a right triangle. The hypotenuse of this triangle is the bottom seam.

We have accurate measurements of one edge and two heights. From these we can compute the others.

```
from myst_nb import glue
```

The measurements include some heights of isoscoles triangles, from which we need to work out the lengths of the edges.

The aft face, for example is 48 inches across the base, w, and 27 tall, h. We can divide this into two right triangles, and compute the remaining edge, e, length.

$$e = \sqrt{\frac{w}{2}^2 + h^2}$$

2.3.1 Computing the six edges

We've measured one edge and two heights:

- The the top of the aft section edge is 48 inches.
- The height of the base is 27 inches.
- The height of the top is 46 inces.

From these, we can compute the remaining edges applying the rule $e^2 = \frac{w}{2}^2 + h^2$. This works because each isoscolese triangle is two right triangles.

```
from sympy import *
a_1, a_2, a_3, a_4, a_5, a_6 = symbols('a_1 a_2 a_3 a_4 a_5 a_6')
```

```
# The "width" (actually the top) of the aft triangle.
a_1 = 48

# The two sides of the aft triangle, given a height of 27".
a_2 = sqrt((Rational(1, 2)*a_1)**2 + S(27)**2)
a_3 = a_2
```

If the top were not truncated, it would be a triangle with a base of 48 inches, and height that's a little more than 46 inches. We'll stick with the truncated measurement of 46 inches, rather than estimate the full, untruncated height.

```
# The two edges of the top, given a height of 46".
a_4 = sqrt((Rational(1, 2)*a_1)**2 + S(46)**2)
a_5 = a_4
```

The seam along the bottom is the hypotenuse of a 46" by 27" right triangle.

```
a_6 = sqrt(S(27)**2 + S(46)**2)
```

We can take the mean of these edges to estimate a regular tetrahedron that has a similar volume.

```
m = (a_1 + a_2 + a_3 + a_4 + a_5 + a_6)/6
```

```
glue("edge", radsimp(m))
glue("edge_f", m.evalf(4))
```

$$8 + \frac{\sqrt{2845 + 6\sqrt{145} + 4\sqrt{673}}}{6}$$

$$46.23$$

This suggests an edge length of $8 + \frac{\sqrt{2845 + 6\sqrt{145} + 4\sqrt{673}}}{6}$, which is approximately 46.23 inches.

The volume needs to be converted from cubic inches to gallons, using the constant $231 \frac{\textbf{cu in}}{\textbf{gal}}$

```
V_r = m**3 / (6 * sqrt(2)) / 231
```

```
var("V")
glue("V_r", Eq(V, radsimp(V_r)))
glue("V_r_f", V_r.evalf(4))
```

$$V = \frac{\sqrt{2}\left(48 + \sqrt{2845 + 6\sqrt{145} + 4\sqrt{673}}\right)^3}{598752}$$

$$50.4$$

This estimates a volume of $V = \frac{\sqrt{2}\left(48 + \sqrt{2845} + 6\sqrt{145} + 4\sqrt{673}\right)^3}{598752}$, which is approximately 50.40 gallons.

This is close to the triangular prism estimate, so this is also a useful guage of the space.

2.3.2 Smallest and Largest

Does it help to bracket this estimate with two other estimates, the least, l, and the greatest, g? If we split these, it's tolerably close to other estimates.

```
l = min([a_1, a_2, a_3, a_4, a_5, a_6])
g = max([a_1, a_2, a_3, a_4, a_5, a_6])
V_l = l**3 / (6 * sqrt(2)) / 231
V_g = g**3 / (6 * sqrt(2)) / 231
```

```
glue("l", l)
glue("g", g)
glue("V_l", V_l)
glue("V_l_f", V_l.evalf(4))
glue("V_g", V_g)
glue("V_g_f", V_g.evalf(4))
glue("V_lg_f", (V_l.evalf(4)+V_g.evalf(4))/2)
```

$$3\sqrt{145}$$

$$\sqrt{2845}$$

$$\frac{435\sqrt{290}}{308}$$

$$24.05$$

$$\frac{2845\sqrt{5690}}{2772}$$

$$77.42$$

$$50.74$$

This varies from $\frac{435\sqrt{290}}{308}$ to $\frac{2845\sqrt{5690}}{2772}$. These extemes are approximately 24.05 gallons to 77.42 gallons. The midpoint, 50.74 gallons, also seems to agree with other estimates.

2.3.3 Truncation

There's a small 9" tetrahedron we could truncate from this volume to improve accuracy. What's its volume?

```
V_t = 9**3 / (6 * sqrt(2)) / 231
```

```
glue("V_t", V_t)
glue("V_t_f", V_t.evalf(4))
```

$$\frac{81\sqrt{2}}{308}$$
$$0.3719$$

This tiny bit of space is $\frac{81\sqrt{2}}{308}$, which is approximately 0.37 gallons. It's negligible, however, being less than a gallon, and we can safely ignore it.

2.4 Tapered Triangular Prism

A regular prism is a solid with two triangular faces joined by three rectangular faces.

Referring back to the diagram, this is clearly not a good description of the space. The aft and and forward end are triangular, but the remaining three faces are not rectangles; the remaining faces are irregular quadrilaterals.

One end of the prism, the aft end, or "base", is an isoscolese triangle, 27" by 48". The other end of the prism, the forward end, is an isoscolese triangle, 8" by 10½". The overall length is 46".

We can use calculus to sum an infinte sequence of triangles from the small forward end to the large aft end. This will compute the volume of the prism.

```
from myst_nb import glue
from sympy import *
```

2.4.1 Approach

The volume, V, of a regular triangular prism is the area of the triangle at either end, a, integrated along the length of the prism, l. Since the area is fixed, $a = \frac{h\times w}{2}$, we have this.

$$V_p = \int_0^l a\mathrm{d}z = \int_0^l \frac{h\times w}{2}\mathrm{d}z = \frac{h\times w\times l}{2}$$

In our case, we don't have the same triangle at each end with a fixed area. Instead, we must define area as a function of the offset from the forward end of the tank, z, $a = A(z)$, leading to a slightly more complex integral.

$$V = \int_0^l A(z)\mathrm{d}z$$

When we compute the area at any point along the fore-to-aft axis of the tank, we can accurately compute the volume within that infinite sequence of triangles.

As background, here's the regular prism volume computation done with `sympy`. This shows how we can translate the math to Python, and use this confirm our results.

```
h, w, l, z = symbols("h w l z")

Integral(h*w/2, (z, 0, l))
```

$$\int\limits_0^l \frac{hw}{2}\,dz$$

```
Integral(h*w/2, (z, 0, l)).doit()
```

$$\frac{hlw}{2}$$

The first step, then, is to compute the area of any triangle from the forward-most $10\frac{1}{2}\times 8$ to the aft-most 48×27.

2.4.2 Measurements

Here are the essential measurements. We've defined these as a dictionary so we can substitute them into other equations. This allows us to change the measurements and get results.

```
var("h_f, h_a, w_f, w_a, l_fa")
measured = {
    # Forward triangle, in inches
    "h_f": 8,
    "w_f": 10 + Rational(1, 2),

    # Aft triangle, in inches
    "h_a": 27,
    "w_a": 48,

    # Overall length from forward to aft, in inches.
    "l_fa": 46,
}
```

Here are the essential width as a function of height, w_h, and area based on height, A_h, given a height of the triangle, h.

```
h, w_h, l, A, z, A_h, A_z = symbols('h w_h l A z A_h A_z')

glue("wfh", Eq(w_h, factor(w_a/h_a * h), evaluate=False))
w_h = w_a/h_a * h

glue("area_h", Eq(A_h, Rational(1, 2) * h * w_h))
```

$$w_h = \frac{hw_a}{h_a}$$

$$A_h = \frac{h^2 w_a}{2h_a}$$

If we assume the fore and aft triangles are congruent, then the width, w, of the triangular face of the prism is a function of height, h.

$$w_h = \frac{hw_a}{h_a} \tag{2.3}$$

Area, $A = \frac{1}{2} \times w \times h$, can then become a function of height, h.

$$A_h = \frac{h^2 w_a}{2h_a} \tag{2.4}$$

Height, h, depends on the distance along the z-distance, meadured from the front of the tank. The $z = 0$ position is the forward edge, with a measured height of h_f. The $z = l_{fa}$ position is length from the forward edge to the aft edge; this has a measured height of h_a.

$$h(z) = \frac{\Delta h}{\Delta z} \times z + h_f = \frac{h_a - h_f}{l_{fa}} \times z + h_f$$

We're assuming each of the two ends of the prism are congruent. This means we're imposing the aft width and height on the tank as a whole.

Is this a safe assumption? Spoiler alert: it isn't.

2.4.3 Challenging the Assumption

We've assumed the shape of the forward and aft end of the tank are congruent triangles. We're treating this as a triangular section that grows in size from a small 8" by 10½" triangle to a large 27" by 48" triangle.

Since $\frac{8}{10\frac{1}{2}} \neq \frac{27}{48}$, we can see the two triangles aren't congruent. This simplifies to $\frac{16}{21} \neq \frac{9}{16}$.

Here are the two triangle area computations based on height. $A_a(h)$ uses the aft height-to-width ratio, $A_f(h)$ uses the forward height-to-width ratio. The $A(h)$ computations are quite different.

```
w_a_h = w_a/h_a * h
w_f_h = w_f/h_f * h

A_a_h = Rational(1, 2) * h * w_a_h
A_f_h = Rational(1, 2) * h * w_f_h
```

```
A_a_h.subs(measured)
```

$$\frac{8h^2}{9}$$

```
A_a_h.subs(measured).evalf(3)
```

$$0.889h^2$$

```
A_f_h.subs(measured)
```

$$\frac{21h^2}{32}$$

```
A_f_h.subs(measured).evalf(3)
```

$$0.656h^2$$

This means a simple $A(h)$ computation using the height, h, to compute the area isn't going to be very useful. We have two choices:

- Define area, $A(z)$ based on independent $h(z)$ and $w(z)$. We can use $A(z) = \frac{h(z) \times w(z)}{2}$.
- Use a midpoint ratio of width to height, r_m, to define area. If $w = r_m \times h(z)$, then $A(z) = \frac{h(z) \times w}{2} = \frac{h(z)^2 \times r_m}{2}$.

Using a midpoint ratio is slightly simpler, but suffers from a problem of being inaccurate. The difference between $0.889h^2$ and $0.656h^2$ for small values of h will be significant.

We need to compute the area using independent $h(z)$ and $w(z)$ functions.

2.4.4 Computing the Area

We can use independent $h(z)$ and $w(z)$ functions to compute the overall area of each triangular section of the tank.

We'll assume these are linear functions of the form $y = mx + b$. The slope, m, for height is $\frac{\Delta h}{\Delta z}$, and the intercept, b, is the height at the forward end, h_f. The width equation is similar.

This leads to two equations for h_z and w_z, which are functions of the distance from the forward end, z.

```
h_z = (h_a - h_f) / l_fa * z + h_f
glue("hfz", h_z)
w_z = (w_a - w_f) / l_fa * z + w_f
glue("wfz", w_z)
```

$$h_f + \frac{z\left(h_a - h_f\right)}{l_{fa}}$$

$$w_f + \frac{z\left(w_a - w_f\right)}{l_{fa}}$$

The height, $h(z) = h_f + \frac{z\left(h_a - h_f\right)}{l_{fa}}$. The width, $w(z) = w_f + \frac{z\left(w_a - w_f\right)}{l_{fa}}$.

```
A_z = factor(expand(Rational(1, 2) * h_z * w_z))
glue("area_z", A_z)
```

$$\frac{\left(h_a z + h_f l_{fa} - h_f z\right)\left(l_{fa} w_f + w_a z - w_f z\right)}{2 l_{fa}^2}$$

From $h(z)$ and $w(z)$, we can compute the area, A_z, as a function of the distance along the Z axis, $A(z) = \frac{1}{2}h(z)w(z) = \frac{\left(h_a z + h_f l_{fa} - h_f z\right)\left(l_{fa} w_f + w_a z - w_f z\right)}{2 l_{fa}^2}$.

This is a bit bulky. We can try to simplify it. First, however, we need to test it to be sure it produces proper area values.

We can evaluate the $h(z)$ and $w(z)$ functions at $z = 0$ and $z = l_{fa}$ to be sure we've got them right. We expect $h(0) = h_f$, $h(l_{fa}) = h_a$, $w(0) = w_f$, and $w(l_{fa}) = w_a$. We can also substitute the actual measurements to compute values for the fore and aft triangles to be sure they match the original measurements.

```
h_z.subs({z: 0}).evalf()
```

$$h_f$$

```
w_z.subs({z: 0}).evalf()
```

$$w_f$$

```
h_z.subs({z: l_fa}).evalf()
```

$$h_a$$

```
w_z.subs({z: l_fa}).evalf()
```

$$w_a$$

```
glue("h_f", h_z.subs(measured).subs({z: 0}).evalf())
```

$$8.0$$

```
glue("w_f", w_z.subs(measured).subs({z: 0}).evalf())
```

$$10.5$$

```
glue("h_a", h_z.subs(measured).subs({z: measured['l_fa']}).evalf())
```

$$27.0$$

```
glue("w_a", w_z.subs(measured).subs({z: measured['l_fa']}).evalf())
```

$$48.0$$

To confirm that we've done this right so far, let's check the model against reality.

At the forward end of the tank, this model predicts a triangle 10.5 across the top, with a height of 8.0. This matches the 10.5×8 actual.

At the aft end of the tank, this model predicts a triangle 48.0 across the top, with a height of 27.0. This matches the 48×27 actual, also.

Now that we can compute the shape of the triangle at each end of the space, we can compute the area, $A(z) = \frac{h(z)w(z)}{2}$. From this, we can then compute the volume.

2.4.5 Volume based on overall length

The volume is the integral of the areas, $A(z)$ where z varies from zero to the length of the prism, l_fa.

$$V = \int_0^{l_{fa}} A(z) \mathrm{d}z$$

For a regular prism this is the $V_p = \frac{hlw}{2}$ formula. Our area is not simply $\frac{hw}{2}$, it's $A(z) = \frac{(h_a z + h_f l_{fa} - h_f z)(l_{fa} w_f + w_a z - w_f z)}{2 l_{fa}^2}$.

```
var("V")
glue("V", Eq(V, Integral(A_z, (z, 0, l_fa))))
```

$$V = \int\limits_{0}^{l_{fa}} \frac{(h_a z + h_f l_{fa} - h_f z)(l_{fa} w_f + w_a z - w_f z)}{2 l_{fa}^2}\, dz$$

The volume is computed with

$$V = \int\limits_{0}^{l_{fa}} \frac{(h_a z + h_f l_{fa} - h_f z)(l_{fa} w_f + w_a z - w_f z)}{2 l_{fa}^2}\, dz \tag{2.5}$$

We can substitute our measurements to get the volume. We'll apply the magical 231 cubic inch per gallon factor to get the volume in gallons of fresh water.

```
V = Integral(A_z.subs(measured), (z, 0, measured['l_fa']))
V_r = (V.doit()/231).limit_denominator(100)
f"{floor(V_r)} {frac(V_r)} gallons"
```

```
'56 79/90 gallons'
```

```
V_r.evalf(3)
```

$$56.9$$

2.4.6 Simplified form

We can create decimal approximations for the fractions, and work with a direct computation that avoids integration. It's not clear that this is simpler. The generic `simplify()` is a poor choice.

```
var("l")
simplify(Integral((A_z/231).evalf(3), (z, 0, l_fa)))
```

$$\frac{0.00216\left(h_f l_{fa}^3 w_f + l_{fa}^3\left(\frac{h_a w_a}{3} - \frac{h_a w_f}{3} - \frac{h_f w_a}{3} + \frac{h_f w_f}{3}\right) + l_{fa}^2\left(\frac{h_a l_{fa} w_f}{2} + \frac{h_f l_{fa} w_a}{2} - h_f l_{fa} w_f\right)\right)}{l_{fa}^2}$$

This variation collects the various factors together, giving a closed form that's kind of workable. It involves terms based on l_{fa}, l_{fa}^2, and l_{fa}^3 which seems about right. This has a single constant term out front for the conversion from cubic inches to gallons.

(Using cubic centimeters and liters would avoid the magical 231 cubic inches per gallon.)

We can try and factor the polynomial , which leads to a much simpler-looking computation of volume.

```
V_c = factor(simplify(Integral((A_z/231).evalf(3), (z, 0, l_fa))))
V_c
```

$$0.000361 l_{fa}\left(2h_a w_a + h_a w_f + h_f w_a + 2h_f w_f\right)$$

This seems to be an elegant closed-form equatio for computing volume from the given measurements. We can recompute the volume as our measurements improve.

```
V_c.subs(measured)
```

$$56.9$$

2.4.7 Matrices

Note that in the closed form volume equation, each term has some combination of h_a, w_a, h_f, and w_f, and unavoidable source of complexity. This "sum-of-combinations" suggests there may is a matrix expression to summarize this complexity.

The following nonsense shows that we can reproduce the volume formula. The following nonsense lacks a clear interpretation. Because it happens to work, it's likely related to the proper scalar triple product (or box product).

```
M_h = Matrix([h_a, h_f])
M_w = Matrix([w_a, w_f])
```

```
V_m = l_fa*(M_h*M_w.transpose()).vec().dot(Matrix([S(1)/3, S(1)/6, S(1)/6, S(1)/3]))/
↪(2*231)
V_m
```

$$\frac{l_{fa}\left(\frac{h_a w_a}{3} + \frac{h_a w_f}{6} + \frac{h_f w_a}{6} + \frac{h_f w_f}{3}\right)}{462}$$

```
V_m.subs(measured).evalf(3)
```

$$56.9$$

2.5 Summary and Conclusion

We've looked at a number of alternatives for estimating the volume of the V-berth fuel tank. Each of these uses some slight simplifications of the geometry to make the math a little simpler.

- **Regular Triangular Prism**. We picked a mid-point between the two triangular faces and used this to estimate the overall size. The resulting volume was 50.97 gallons.
- **Regular Tetrahedron**. We used two approaches here.
  - We picked a mid-point among the various edges and used this to estimate the size. The resulting volume was 50.4 gallons.
  - We used the largest and smallest edges to compute two sizes and took the mid-point. The resulting volume was 50.74 gallons.
- **Irregular Triangular Prism**. The tank is a prism that tapers from aft to forward. We can describe this taper as a function of the distance along the fore-and-aft axis of the tank. The resulting value was 56.9 gallons.

The simplifying assumption of congruent triangles at each end of the tank is implicit in the initial methods. This is exposed by the irregular prism computation, where we computed areas and volumes with fewer simplifying assumptions.

The use of *sympy* makes it easy to perform the algebraic work and then confirm the formula with measured values.