

The Phantom of the Opera^{tions}

Dianne P. O’Leary

Suppose you work for the Food and Drug Administration (FDA). A manufacturer wants approval for a new computerized tomography (CT or CAT) machine that is claimed to be better than a currently used machine at detecting lung cancer. How can you tell whether the claim is true?

Ideally, we would like to take a set of patients, use both machines to scan for lung cancer, wait 6 months, and then see which machine gave a better answer. This is not practical; it is not even ethical to use both machines on a single patient, because that gives an X-ray dose higher than necessary and risks harm to the patient. Similar experiments with animals also raise ethical questions.

The common alternative is to use both machines on a **phantom**, an object with characteristics (size, shape, density, etc.) similar to typical lung cancer tumors. The phantom is positioned inside a model of the human chest. The machine that can more closely estimate the phantom’s location and size (volume) is said to be “better”.

To assess this, we first need a very accurate estimate of the volume of the phantom. So the goal of this project is to determine how to compute a reliable estimate, with error bounds, of the volume of an object that is tumor-like.

The “gold-standard” way to measure volume is to use Archimedes’ principle. Take a very precisely calibrated flask (i.e., a measuring cup), add a known volume of water, drop the solid object in, and measure the resulting volume. The change in volume is the volume of the object.

Unfortunately, in order to better match the characteristics of a tumor, the phantom is porous and absorbs water, so the gold-standard cannot be applied. Instead, what we have are a set of rather precise coordinates of points on the surface of the phantom, with uncertainties, and from that we need to estimate the volume, knowing that the phantom is close to spherical in shape.

To make things more difficult, the points are not at all equally distributed on the surface, because the measurements were made by clamping the phantom to hold it steady and then using a laser to measure at convenient points given the clamped position. It was difficult to measure points near the “north and south poles”; most measurements are close to the “equator”.

I have the data for two phantoms measured at the National Institute of Stan-

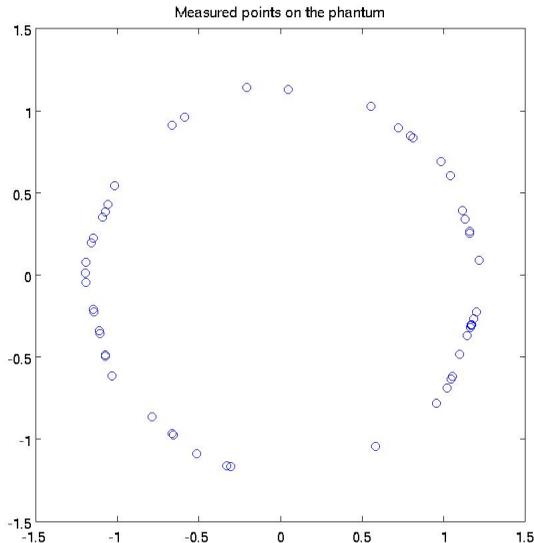


Figure 1. *The 50 measured points for the phantom.*

dards and Technology (NIST), courtesy of David Gilsinn,¹ so we could estimate the volumes and compare with his estimates. But 3-D datafitting is rather difficult to program, so for the purposes of this case study, I’ve artificially generated 50 points on the surface of a 2-D phantom that is close to circular in shape, and we will use that “phantom phantom” in our study.

You can find the coordinates \mathbf{x}_i , $i = 1, \dots, M$, with $M = 50$, in the file `measuredpts`, and they are shown in Figure 1. The error in each measured value is normally distributed with mean 0 and standard deviation 10^{-4} .

We start by determining how close to circular the phantom is, resulting in our first estimates of its area.

CHALLENGE 0.1. Use the coordinates \mathbf{x}_i , $i = 1, \dots, M$, to estimate the center of the phantom and the radius r . For a circle, the center point $\mathbf{c} = [c_1, c_2]$ has the property that its distance from any point equals the radius, so let’s estimate the “center” of the phantom by minimizing

$$\sum_{i=1}^M (\|\mathbf{x}_i - \mathbf{c}\|^2 - r^2)^2$$

¹David E. Gilsinn, Bruce R. Borchardt, and Amelia Tebbe, “Estimating Volumes of Near-Spherical Molded Artifacts,” *Journal of Research of the National Institute of Standards and Technology* 115(3), 149–177 (2010) <http://nvl.nist.gov/pub/nistpubs/jres/115/3/V115.N03.A01.pdf>.

over all choices of $r > 0$ and \mathbf{c} . This is a **nonlinear least squares problem** that can be solved using MATLAB's `lsqnonlin`.²

- Report your estimates for \mathbf{c} and r .
 - Make a quantitative statement of how close to circular the phantom is.
 - Using the largest and smallest distances $d_i = \|\mathbf{x}_i - \mathbf{c}\|$, give upper and lower bounds on what you think the area of the phantom is.
-

Next, we will use our 50 measured points to generate an approximation to the boundary of the phantom.

Imagine drawing a line segment at angle θ , starting from \mathbf{c} and ending at the boundary of the phantom. Let $d(\theta)$, for $0 \leq \theta \leq 2\pi$, be the length of that line segment.³

CHALLENGE 0.2.

- (a) For each data point \mathbf{x}_i , determine the corresponding angle θ_i . Fit an interpolating cubic spline to the datapoints (θ_i, d_i) , $i = 1, \dots, M$ to obtain an approximation $\hat{d}(\theta)$. (Set up the coordinate system so that the first datapoint corresponds to $\theta = 0$, and repeat this datapoint at $\theta = 2\pi$ to make the function 2π periodic.)
 - (b) Graph the 50 data points as circles, and graph the function $\hat{d}(\theta)$ (evaluated at 500 equally-spaced points) as a curve.
 - (c) The function $\hat{d}(\theta)$ can be used to define an approximation to the boundary $b(\theta)$ of the phantom. Graph the 50 datapoints \mathbf{x}_i as circles, and graph the boundary approximation $\hat{b}(\theta)$ (evaluated at 500 equally-spaced points) as a curve.
-

Having a definition for \hat{b} gives us many ways to estimate the area. We will experiment with three ways.

²Later in the course we'll talk about how `lsqnonlin` works, but for now we can read MATLAB's documentation and use `lsqnonlin` as a "black box" without worrying about what algorithm it uses.

³If any line through \mathbf{c} intersects the boundary of the phantom exactly twice, once on each side of \mathbf{c} , then $d(\theta)$ is a well-defined function. This is true for a circle, and we assume it for our near-circular phantom.

POINTER 0.1. Stokes Theorem

The Stokes Theorem (or, in 3-D, the Divergence Theorem) relates an integral over the interior of a region to an integral over the boundary of the region. We can use it to estimate the area of the phantom. The resulting formula for the area is

$$\frac{1}{2} \int_0^{2\pi} \mathbf{n}(\theta)^T \mathbf{x}(\theta) d\theta,$$

where $\mathbf{x}(\theta)$ is a point on the boundary. The vector $\mathbf{n}(\theta)$ is the **outward normal**, a vector orthogonal to the boundary at θ , pointing out from the region and normalized to have length 1.

CHALLENGE 0.3.

- (a) Use Monte Carlo integration with 1000, 5000, and 10000 points to estimate the area enclosed by $\hat{b}(\theta)$. Estimate the error for each of the estimates.
 - (b) Use MATLAB's `quad` function to estimate the area by partitioning the phantom into 2 halves and summing the area that `quad` estimates for each.
 - (c) Use the Stokes Theorem and `quad` to estimate the area.
-

We obtained \hat{d} and \hat{b} by **interpolating** our datapoints. But the datapoints had measurement error, so this might not be the right strategy. Instead, we might seek smoother functions \tilde{d} and \tilde{b} that are close to the measured values but do not necessarily pass through them. We'll repeat our experiments with a smoother spline.

CHALLENGE 0.4. MATLAB has several functions in the Spline Toolbox for approximating datapoints using a spline. Choose one of `spap2`, `spaps`, or `csaps`, document any choices you make for parameters, and use the MATLAB function to determine \tilde{d} . Plot \hat{d} and \tilde{d} and compare them. Then repeat the previous challenge using \tilde{d} and the resulting \tilde{b} .

Now that you have many estimates of the area, it is time to commit to a single answer to report for the area of the phantom.

CHALLENGE 0.5.

- (a) Make a table containing all of your area estimates, your corresponding estimate of the errors, and your estimates of the number of function evaluations used to obtain the estimates.
 - (b) Which method do you think gave the best estimate? Why?
 - (c) List all the sources of error that you can think of, in order of importance.
-