Wave Slicing: Exploring Periodic Functions

The sun rises reliably in the east. Spring follows Winter, every time. Hearts beat, lungs expand and contract, car wheels spin, and sirens wail. The school day and school year follow familiar, repeating rhythms.

With all this repetition in the world, it makes sense that students should study periodic relationships. In school mathematics, however, "periodic" often means "trigonometric." Does that mean you have to take trig to explore periodic phenomena? No. In the course of an NSF-sponsored curriculum development project,[1] we came across an alternative we call *wave slicing*. In this article, we'll see how students can determine the shape and period of periodic data by slicing it up and superimposing the slices. Our strategy uses the "modulo" or "remainder" function common in data analysis packages. The idea is straightforward, and springs directly from the fundamental requirement for a periodic function: $f(t) = f(t + P)$.

Here's how to think about it: periodic data repeats with some period. Let's call that period P. If the data really repeat, we could take the data and chop it up into segments that are P long, and superimpose them. If we do it right, the different periods should all line up. Figure 1 shows schematically how this works.

---

1 Award number DMI-0216656, adapting a technique from Erickson (2001).

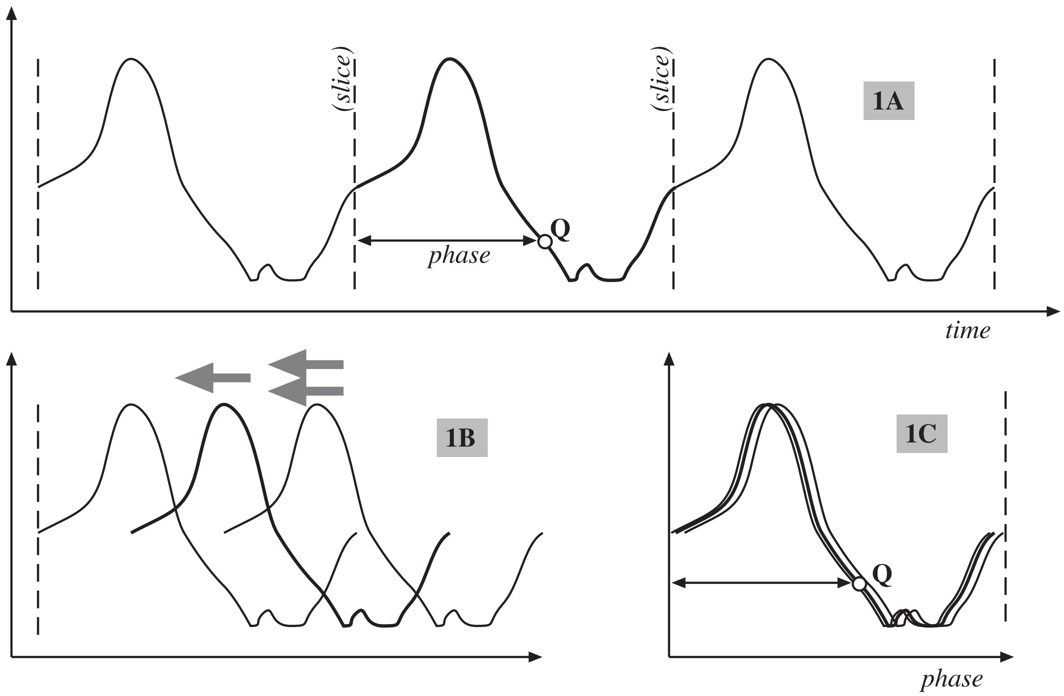

Figure 1. The main idea: Slice the data into segments one period long (1A) and superimpose them (1B and 1C).

This simple idea generates two problems: how to chop up the data and superimpose it, and how to find *P*. The short answers are, students can use the modulo function to do the chopping, and the resulting plot to find *P* with amazing precision. The rest of this article fills in details and provides examples.

Slicing with Modulo

Figure 1 illustrates this slicing-and-superimposing idea. What's the difference between graphs 1A and 1C in Figure 1? The vertical axes—the signal, whatever it is—are the same. But the horizontal axis in 1A is *time*, while in 1C, the horizontal position of any point is really a *phase*: the amount of time each point is into its own current cycle.

Point Q in Figure 1 is an example. When we superimpose the cycles, its horizontal coordinate is not its original time coordinate. Its phase is the time minus the time of the beginning of its cycle.

We could calculate this phase (let's call it ϕ) using division and the greatest-integer function. If t is the time and P is the period, the number of complete cycles is $\mathrm{int}(t/P)$. The starting time of the "current" cycle is therefore $P \times \mathrm{int}(t/P)$, and ϕ is:

$$\phi = t - [P \times \mathrm{int}(t/P)] \qquad (1)$$

A student can therefore estimate the period P, then calculate ϕ for each value of t, and finally plot the original signal against ϕ to get the "superimposed" graph of Figure 1C. This is still pretty complicated for many students. There are many ways equation (1) can go wrong, depending on the particular syntax of the data analysis tool a student is using.

A different function, however, makes it easier: *modulo*. This function, common in computer languages and data analysis packages, takes two arguments, returning the *remainder* when the first is divided by the second. For example, modulo(23, 10) = 3, (or, commonly, 23 mod 10 = 3).

Using that function, equation (1) becomes much simpler:

$$\phi = \mathrm{modulo}(\ t, P\ ). \qquad (2)$$

Students remember *remainder*, so that is a good way to talk about it. The change from their elementary days is that t and P don't have to be whole numbers, but that doesn't matter. The remainder still removes as many whole periods P from t as possible, and tells you what's left.

Using Real Data

Let's see how this works when we use real data. Figure 2 shows sound amplitude as a function of time.[2] We want to know the pitch of the sound, so we need the period (or the frequency) of the wave. If we were doing this "professionally," we would pipe the signal into some signal analyzer or do a Fourier transform. But conceptually, these are black boxes; we would rather students figure out the pitch using tools they understand.

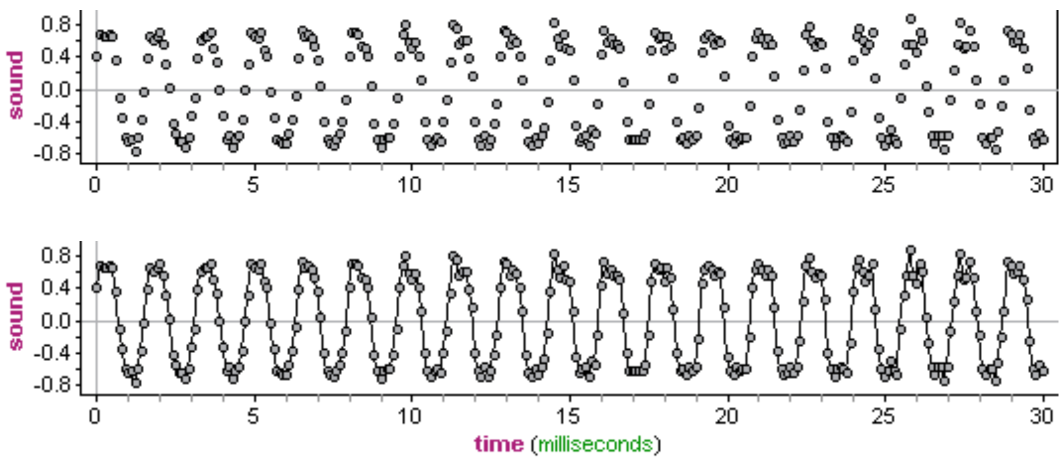

Figure 2 *The sound of air blowing over a test tube. The data (identical in both graphs) are quite sparse—only a dozen or so points per cycle. Connecting the dots in the lower illustration makes it clearer that* **sound** *is periodic.*

Figure 2 is a screen shot from Fathom™, a school data analysis package (Finzer 2007). We could just as well use another package (such as Excel, Logger *Pro*, or Tinker-

2 The author blew across the top of a test tube, recording and digitizing the sound using a Vernier microphone.

Plots). We could even use a graphing calculator, though the computer's speed and large high-resolution display make this analysis much easier.

Whatever tools we use, our data are stored in two variables, **time** and **sound**. To make wave-slicing work, we need a new variable, **phase**. In Fathom, we give it the formula **phase = modulo(time, period)**. (In Excel, the function is named **MOD** and works the same way.) We need a value for **period**. We can estimate it from the graph: it looks like about 1.5 milliseconds.

But what is the *best* value for **period**? We will find it using the *sound*-*phase* graph. If we vary **period**, the computer will recalculate all the **phase**s. That is, it re-overlaps all the waves. If **period** is wrong, the waves won't line up. When they do, the value of **period** is correct.

Seeing how the *sound*-*phase* graphs change as you change **period** is striking. The graphs in Figure 3 give you some idea of what this looks like. Seeing them change dynamically is more visceral; you can make such a dynamic plot easily in Fathom using a "slider" for **period** (or, with a little more work, in Excel by using a scroll bar).

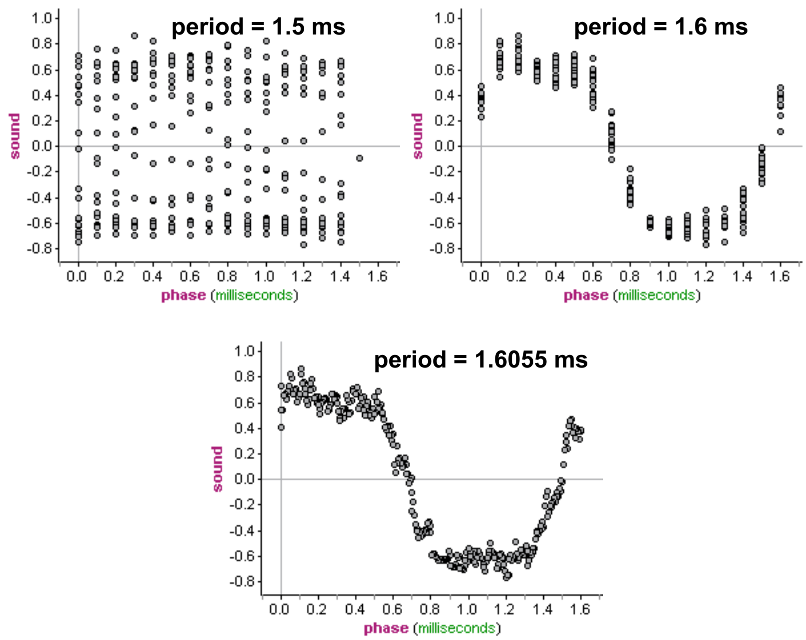

Figure 3 *Refining the value of* **period***. The first graph shows the* **sound-phase** *graph for* **period** *= 1.5 ms. Fiddling with the value gives the second graph (1.6 ms), and further refinement gives the final graph (***period** *= 1.6055 ms).*

Once the waves line up, we know the period with considerable precision. While our estimate is 1.6055 ms, plausible values range only from about 1.602 to 1.607. Outside that range, you can see that the cycles do not line up properly. This is especially obvious if the graph updates as you change the proposed period. (If you need the frequency, just calculate **1/period**. In this case, 1/1.6055 ms is 623 Hz.)

Figure 4 shows the end point of our fit in Excel, and also illustrates how the formulas work if you use that program.

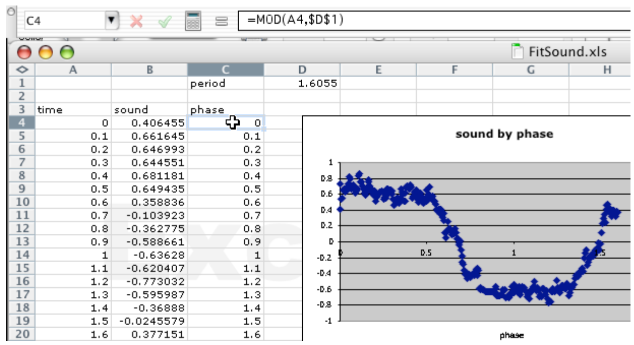

Figure 4 *Showing* ***period*** *= 1.6055 ms in Excel.*

Discussion and Refinements

If we wanted students to find the period (or frequency) of our wave, and did not do wave slicing, what would we do? Very likely, we would try to fit a sine to the data. To fit the sine in this case, though, students must juggle four parameters (zero point, amplitude, period, and phase), while here we have just one parameter: the period.

Furthermore, when we superimpose many cycles, the wave “fills in” so we can see its whole shape. Even though it is not obvious looking at a few cycles of the original graph, we can see in the phase graph that the wave is not particularly sinusoidal, just periodic.

There are refinements to this technique. Here are two tips learned though experience:

1. It helps if the “time” that you feed the modulo function begins near zero. For example, if you’re looking at data that begins in 1990, you might use **phase** = **modulo**((**date** – 1990), **period**).
2. To tell whether **period** needs fine adjustment, it is easiest to look at the steep parts of the phase plot.

Extensions

Once you can easily find the period or frequency of a data set, there are many directions you can go. Here is one idea:

- Have a class musician bring in his or her instrument. Each group records a different note.
- Each group determines the frequency of its note and posts it for sharing.
- The class now enters the shared frequency data, and plots frequency against the number of half steps their note is above the lowest note.
- Now they try to fit the function; it is exponential with base $2^{(1/12)}$, or about 1.0595.

Reflection

We don't teach the modulo function in school. However:

1. This technique is closer, conceptually, to the core of periodic data: that the data repeat. Many students justifiably wonder what right triangles and SOH-CAH-TOA have to do with situations like these.
2. Although students may not be familiar with modulo, explaining it in terms of *remainder* makes sense. Students learned about remainder when they first studied division in grade school; meeting a situation where it is relevant again may help them better understand division.
3. The modulo function is not obscure in the real world. It is commonly used in all sorts of data processing applications.

4. Many of us (of a certain age) met this function as “new math” students studying modular arithmetic. Though that reform never took hold, modular arithmetic is intrinsically interesting, and helps us learn important principles underlying mathematical systems.

Why have we not used this technique before? Because it is pointless without 21st-century technology. This is a good example of using technology to enhance conceptual understanding—in this case, of the properties of periodic functions. It is also a good example of how a tool might impact our choices of how—and what—to teach.

Finally, this process gives students a setting in which they can clearly see what a *parameter* is. **Phase** is an “ordinary” data variable—it takes on many different values and you put it on an axis. But **period** is a parameter—a single number that influences the whole analysis.

References

Erickson, Tim. *Data in Depth*. Emeryville, CA: Key Curriculum Press, 2001.

Finzer, William. *Fathom Dynamic Data™ Software*, Key Curriculum Press, 1998, 2007. **http://www.keypress.com/fathom**.

Vernier Software and Technologies. *Logger* Pro *Software*, version 3.4. Beaverton, Oregon: Vernier Software and Technologies, 2007. **http://www.vernier.com**.

[What follows is a Technology “box.” It could go earlier in the article. I think its figure captions are superfluous.]

Hooking Up the Microphone and Getting Data

There are many probeware setups—combinations of software and hardware designed for automated data collection—you can use to get the data into your computer. We used Fathom 2.1 (Finzer 2007), a LabPro interface (Vernier Software and Technologies 2007), and a Vernier microphone. Here are instructions for our setup:

- Plug in the LabPro so it turns on; connect it to the computer using the USB cable that comes with it.
- Plug the microphone into one of the ports on the LabPro.
- In Fathom 2.1, drag a new "Meter" off the shelf. A meter appears, showing the current sound level. (If this does not happen automatically, you may need to specify what kind of sensor it is: **Microphone**, as opposed to **Force**, **pH**, etc.)
- Drag a new collection off the shelf.

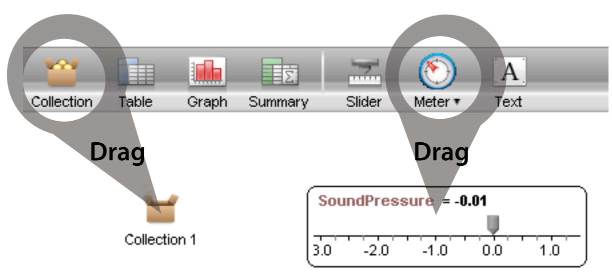

Figure B1 Dragging the meter and collection off the shelf.

- Drag a plug from the meter to the collection. Its name changes to **Experiment with SoundPressure**. The collection's inspector appears, showing that by default you will collect 0.03 seconds of data at 10,000 Hz. That's fine. Note the **Turn Experiment On** button.

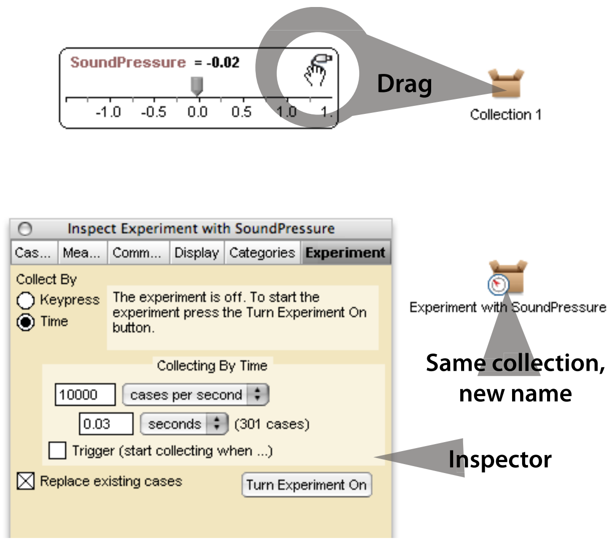

Figure B2 *Connecting the meter to the collection; what the experiment and its inspector should look like..*

- Hold the mike up where it can hear you.
- Start whistling or humming.
- Press **Turn Experiment On**.

And you're done! (It only lasted 0.03 seconds, after all.) You can now make a graph of **sound** against **time** just as we did for Figure 2.