A New and Wonderful Pendulum Period Equation

Tom Van Baak, tvb@LeapSecond.com

Introduction

I thought it would be useful to introduce the pendulum clock community to a very recent development involving mathematics and pendulums. We have all seen equations to compute pendulum period; they have been around, unchanged, for many centuries. Amazingly, just a few years ago a completely new formula was discovered. Even though it won't change the operation or the understanding of any of our precision pendulum clocks, it is still fascinating and useful.

To put the new formula in context let me briefly summarize the existing methods we already know about, with an emphasis on accuracy.

The first pendulum period formula

Every textbook has this formula:

$$T \approx 2\pi\sqrt{\frac{L}{g}}$$

Honest authors will use ≈ (approximately equal) to hint that the formula is not exact. Students are sometimes taught the "formula is valid for small angles", without being told just what *valid* or *small* specifically mean.

If one wants to use = (exactly equal) the formula can be written as:

$$T = 2\pi\sqrt{\frac{L}{g}}\left(1 + CE\right)$$

In this form, CE is some small magic circular error correction term.

The second pendulum period formula

Circular error is, of course, a function of amplitude, θ. In hundreds of books and articles, advanced physics or horological texts are quick to state, or even derive, the circular error correction term. A particularly good example is the NAWCC or HSN paper by George Feinstein, "Impulsing the Pendulum: Escapement Error".

I have limited experience myself with differential equations and elliptic integrals, but I can appreciate the beauty of the resulting formula:

$$T \approx 2\pi \cdot \sqrt{\frac{L}{g}} \cdot \left(1 + \left(\frac{1}{2}\right)^2 \sin^2\frac{\theta}{2} + \left(\frac{1\cdot 3}{2\cdot 4}\right)^2 \sin^4\frac{\theta}{2} + \left(\frac{1\cdot 3\cdot 5}{2\cdot 4\cdot 6}\right)^2 \sin^6\frac{\theta}{2} + \left(\frac{1\cdot 3\cdot 5\cdot 7}{2\cdot 4\cdot 6\cdot 8}\right)^2 \sin^8\frac{\theta}{2} + \ldots\right)$$

This is still not a perfectly *exact* formula in the mathematical sense because there are an infinite number of terms. However, it is a perfectly *useful* formula in the engineering sense because it can be made as accurate as you want, by incorporating more and more terms. The pattern of coefficients is clear; all you need is a calculator with **sine** and you can compute period as accurate as you need.

This begs the question, how many terms do you need; what accuracy do you need? And what is the angle; is there a best case and worse case angle? These questions are important because the larger the angle and the greater the precision, the more terms you need to calculate.

This problem is worse if writing computer code and you are not sure ahead of time what angle the user of your program will choose. You may only expect precision pendulum angles like 1 or 2 degrees amplitude. But if someone comes along and plugs in 10 or 30 or 90 degrees, then your answers may be quite wrong.

In many texts, the formula takes another familiar form; the result of multiplying, squaring and removing common factors:

$$T \approx 2\pi \cdot \sqrt{\frac{L}{g}} \cdot \left(1 + \frac{1}{4}\sin^2\frac{\theta}{2} + \frac{9}{64}\sin^4\frac{\theta}{2} + \frac{25}{256}\sin^6\frac{\theta}{2} + \frac{1225}{16384}\sin^8\frac{\theta}{2} + \ldots\right)$$

This formula appears simpler; with denominators always a power of 2, but with more mysterious looking numerators.

The third pendulum period formula

Both formulae above are based on powers of $\sin^2(\theta/2)$ which is not always convenient. So a third version of the formula that we often see is based on θ^2 rather than on $\sin^2$.

$$T \approx 2\pi \cdot \sqrt{\frac{L}{g}} \cdot \left(1 + \frac{1}{16}\theta^2 + \frac{11}{3072}\theta^4 + \frac{173}{737280}\theta^6 + \frac{22931}{1321205760}\theta^8 + \ldots\right)$$

This simplified version works with calculators without a sine function and is the version most commonly appearing in computer programs or Excel macros. The downside is that now both the numerator and denominator are weird numbers; numbers you can only look-up in books or online. When using calculators, note θ is amplitude in radians, amplitude being the angle of deflection from center vertical, which is half the angle of the entire swing, which itself is half the motion of a full period.

A pleasant side effect of these numbers is that you can google for them to find articles that mention circular error or computer code related to pendulums. For example, search for the words – pendulum 173 737280 – and enjoy several pages of fine technical articles. Or search for just the two numbers – 22931 1321205760 – and be surprised.

One thing was strange; Feinstein's article did not always appear in search results. It turns out George wrote 519/2,211,840 for the third coefficient. But 519 and 2211840 are both divisible by 3 which is why you usually see the reduced fraction 173/737280 instead.

Closed form solutions

There are two features that make all variations of existing pendulum period equations awkward. First, you have to calculate or look-up the coefficients. Second, you have to know ahead of time how many terms to use in order to achieve your desired precision. You have to be aware that issues of precision even exist.

When you use **sqrt** you don't think about precision. The output is as precise as the input. When you press **sin** or **cos** or **tan** on a calculator you don't think about precision. You know the result will be correct, safe from issues of approximation or rounding. But pendulum period calculations have never been that simple or safe.

One thing that might bother you late at night is why the area of a circle is simply and exactly πr^2 but the period of an ideal pendulum is neither simple nor exact. Almost every equation we learn in school is simple and exact. When a solution can be expressed as an equation based on basic operations and well-defined functions, mathematicians call it a *closed form* solution. Having an infinite series of strange coefficients is still a solution, but it not closed form.

Now this makes little difference from an engineering perspective, but there's something deeply troubling about such equations from a mathematics or aesthetic perspective, even if they are beautiful in their complexity.

The area of a circle is exactly πr^2, not some infinite series. We are all so used to the pendulum period equation being an exception to the rule that no one thinks about it anymore. But mathematicians always prefer a closed form solution to one with infinite strange terms. It's just that since time began no one could come up with a closed form solution to the pendulum period equation. Until now…

The new pendulum period formula

In 2008, two mathematicians came up with a stunning, simple, closed form expression for pendulum period. The equation is exact; that is, it works perfectly, for all angles. The equation is closed form; that is, there are no strange numbers or series of terms to evaluate. It is based on AGM, the Arithmetic-Geometric Mean.

Note that all the other formulae for period are of the form:

$T \approx 2\pi\sqrt{\frac{L}{g}}\left(1+CE\right)$, where CE = *some* ***complicated*****,** ***infinite*** *series that you have to look up!*

Here is the **new formula for pendulum period:**

$T = 2\pi\sqrt{\frac{L}{g}}\left(1/CE\right)$, where CE = $AGM(1, \cos(\theta/2))$

That's it. No large integers, no sines to the 6th power, no infinite series, no inaccuracy. It's simple, amazing, and perfect. Ok, but what's that AGM function? I don't see it on my calculator.

What is AGM (Arithmetic-Geometric Mean)

Like many amazing mathematical insights, the history of AGM traces back to Carl Friedrich Gauss; around the year 1800. It is sometimes called "Gauss's AGM". In my experience, AGM is a rather obscure function, but it gained significant attention a few decades ago when it was used as a radically new and efficient way to compute π to billions and trillions of digits.

We all know the average, or arithmetic mean (AM) of *a* and *b* is simply $(a+b)/2$. Slightly less well known, but equally simple, the geometric mean (GM) of *a* and *b* is $\sqrt{(a \times b)}$. Note in both cases the resulting mean is a number that is *between a* and *b*.

The Arithmetic-Geometric Mean, or AGM, is a recursive mix of these two means. You start with *a* and *b* and compute their AM and GM. Then you take those two means and compute their AM and GM. And repeat. Within a few steps it converges to a single number, which is called the AGM of *a* and *b*. That's all there is to it.

For example, to compute agm(1, 2) takes only 4 steps. Watch AM and GM **converge**:

0:	1.000000000000000	2.000000000000000
1:	**1.**500000000000000	**1.4**14213562373095
2:	**1.45**7106781186548	**1.456**475315121970
3:	**1.4567910**48154259	**1.4567910**13939555
agm:	**1.456791031046907**	**1.456791031046907**

The numbers don't have to be integers. Here's agm(12.345, 98.765):

0:	12.345000000000001	98.765000000000001
1:	55.555000000000000	34.917816727281220
2:	45.236408363640606	44.043833941700719
3:	44.640121152670659	44.636138476431270
4:	44.638129814550965	44.638129770133474
agm:	44.638129792342220	44.638129792342220

Thus **agm** is an easily calculated mathematical function, not unlike x^2 or √x, sin or cos, exp or log. If it were more well-known or needed on a daily basis it would surely have its own calculator key and Excel function. It’s like a *mean* mean, a perfect *shuttle diplomacy* compromise between an arithmetic mean and a geometric mean.

In summary, start with *a* and *b*, compute new $a = (a+g)/2$ and new $g = \sqrt{a \times g}$, and repeat a few times until a = g or a ≈ g. It converges quadratically, meaning you double the number of digits of accuracy each step. Few algorithms in mathematics are so simple and so fast.

The test program (agm.exe) and source (agm.c) are available (www.leapsecond.com/tools/).

Source code for agm function

For those of you who program, the following is my C code for agm().

```
// Return arithmetic-geometric mean (positive arguments).
double agm (double a, double g)
{
    if (a <= 0 || g <= 0) {    // domain error
```

```
        return 0;
    }
    for (;;) {
        double a0 = a;
        double g0 = g;
        a = (a0 + g0) / 2;
        g = sqrt(a0 * g0);
        if (a == g || fabs(a - g) >= fabs(a0 - g0)) {
            return a;
        }
    }
}
```

This code is available at http://leapsecond.com/tools/agm.h

Note that some implementations of agm found on the web are sub-optimal because they check for a hard-coded epsilon value (or ULP). This version avoids that problem by continuing while the current estimate is more accurate than the previous one.

Source code for pendulum period equation

Once you have the agm function, computing pendulum period can be done in *one line of code*:

```
// Calculate pendulum period (seconds) given amplitude (radians).
double period_rad (double L, double g, double theta)
{
    return 2 * PI * sqrt(L / g) / agm(1, cos(theta / 2));
}
```

If the angle is given in degrees:

```
// Calculate pendulum period (seconds) given amplitude (degrees).
double period_deg (double L, double g, double degrees)
{
    return 2 * PI * sqrt(L / g) / agm(1, cos(PI * degrees / 360));
}
```

For those of you using calculators or Excel and small angles and not worried about ppm or ppb accuracy, it is probably best to stick with any old pendulum period formula you are currently using.

But as you can see, the new formula is very simple to implement in any programming language. And there are no longer any worries about precision, about large or small angles, about how many terms to include in the calculation. The new formula is as exact as possible on the calculator or computer you are using (e.g., single- or double-precision computer arithmetic).

Test results

I wrote a program to compare 8 different versions of the pendulum period formula. For 1 degree amplitude even the plain $2\pi\sqrt{(L/g)}$ formula has an error less than 20 ppm. The other methods give "perfect" results (the accuracy of double precision floating point is about 16 digits).

But if the amplitude is, say, 10 degrees the methods begin to differ in accuracy:

```
2.006373489105913  pendulum_period_simple()   relative error -1.904e-003
2.010200020332652  pendulum_period_sine3()    relative error -2.500e-010
2.010200020835199  pendulum_period_sine6()    relative error -2.209e-016
2.010200020835200  pendulum_period_sine9()    relative error  0.000e+000
2.010200007497852  pendulum_period_taylor2()  relative error -6.635e-009
2.010200020835127  pendulum_period_taylor4()  relative error -3.623e-014
2.010200020835200  pendulum_period_taylor6()  relative error  0.000e+000
2.010200020835200  pendulum_period_agm()      relative error  0.000e+000
```

And if the amplitude is, say, 90 degrees, all except the new AGM formula are horrible:

```
2.006373489105913  pendulum_period_simple()   relative error -1.528e-001
2.352198607101219  pendulum_period_sine3()    relative error -6.758e-003
2.366966943904538  pendulum_period_sine6()    relative error -5.224e-004
2.368091787432353  pendulum_period_sine9()    relative error -4.742e-005
2.359519885736668  pendulum_period_taylor2()  relative error -3.667e-003
2.367882619608746  pendulum_period_taylor4()  relative error -1.357e-004
2.368189700265774  pendulum_period_taylor6()  relative error -6.075e-006
2.368204085981288  pendulum_period_agm()      relative error  0.000e+000
```

You can see why this new formula is of great interest to those who work with large amplitude pendulum calculations.

The program (pend13.exe) and source (pend13.c) are available (www.leapsecond.com/tools/).

A calculator with agm key

I mentioned the agm function is really no more mysterious than functions like sqrt, exp, and log, or sin, cos, and tan.

If there was more use for agm, I'm sure scientific calculators and programming languages would have this function built-in.

Computing agm by hand is as easy as computing sqrt by hand (think: divide and conquer); and much easier than computing sin, cos, or tan by hand. The algorithm is trivial and accuracy grows far quicker than almost any other math function.

Perhaps some day calculators will have an agm key. With the help of a photo editor I upgraded a vintage hp 35 scientific calculator. Can you spot the new agm key?

It would be simple to add an agm key to any calculator program (PC, web, app). Perhaps I will try that.

Motivation and application

I learned about the new AGM-based pendulum period formula while trying to debug a computer program I was writing. It helped that I was familiar with AGM prior to this, due to my long-time interest in digits of pi. If you think the history of precision pendulum clocks is interesting you should read about the even longer history of computation of π sometime! But I digress.

We come to the field of precision pendulum timekeeping from many different backgrounds. We pursue deeper understanding of how precise pendulums operate with many different skills. Some are historians, with vast libraries and deep knowledge of past successes and failures. Some are mathematicians who spin equations with ease. Some are adept at mechanical design and, putting the rest of us to shame, actually construct superb pendulum clocks. Others delight in precise measurement and analysis. So there's equal room for *historical*, *analytical*, and *empirical* methods to studying pendulum clocks. I'd like to add one more: the ***virtual*** method.

As a software engineer I found it more fun to build a virtual pendulum simulator in a home computer than to build a real pendulum in a home machine shop. So it was with some surprise that I discovered my simulations gave results that did not match what the pendulum period formula predicted. Some web research on large amplitude pendulum errors pointed to the AGM paper. In the end, to my delight, I found that my precise simulation of a swinging pendulum was far more accurate than any of the pendulum period equations I was using at the time.

Conclusion

I now use the AGM formula for all my pendulum software; the programs work just as perfectly at 179.99 degree amplitude as they do at 1 degree. I no longer have to write code with strange integers or worry about how many terms I should use to obtain precise results.

Ever since the pendulum clock was invented, all of us must have assumed that the period equation must necessarily be complex and only approximate. The new formula now puts pendulum period in the same category as the area of a circle. The closed form AGM-based pendulum period equation gives the exact answer for an ideal pendulum of any amplitude, just like πr^2 gives the exact area for a circle of any size.

I realize any practical person with a 1 degree pendulum needs little more than one or two terms of circular error correction in a period calculation. But for those of us looking for impossibly high precision, or looking for perfection, or playing with computer simulations, this new formula is really amazing. The fact that it remained undiscovered all this time adds to the excitement.

After centuries of waiting we can now write a pendulum period equation with = instead of ≈.

$$T = \frac{2\pi}{agm(1, \cos(\theta/2))}\sqrt{\frac{L}{g}}$$

Further reading

- Here's the original 2008 paper that introduced the new pendulum period formula:

  Approximations for the period of the simple pendulum based on the arithmetic-geometric mean
  http://suppes-corpus.stanford.edu/articles/physics/431.pdf
  by Claudio G. Carvalhaes, Patrick Suppes

- This is a copy of the 2006 Feinstein NAWCC article (it also appeared in HSN 2005-3):

  Impulsing the Pendulum: Escapement Error
  http://www.nawcc-index.net/Articles/Feinstein-ImpulsingThePendulum.pdf
  by Dr. George Feinstein

- Good introduction to elliptic integrals, AGM, and pendulums:

  An Eloquent Formula for the Perimeter of an Ellipse
  http://www.ams.org/notices/201208/rtx120801094p.pdf
  by Semjon Adlaj, 2012

- This article nicely explains what is meant by "closed form". If you google for words like – pendulum period closed form – you will find many more like it:

  Closed Forms: What They Are and Why We Care
  http://www.ams.org/notices/201301/rnoti-p50.pdf
  by Jonathan M. Borwein, Richard E. Crandall

- Part 1 of an excellent series of articles on AGM, and its relation to PI:

  Arithmetic-Geometric Mean of Gauss
  http://paramanands.blogspot.com/2009/08/arithmetic-geometric-mean-of-gauss.html
  by Paramanand Singh, 2009

- More on elliptic integrals:

  The AGM Simple Pendulum
  http://arxiv.org/pdf/1202.2782.pdf
  Mark B. Villarino, 2012

- This paper wins my prize for carrying out the period equation to most terms!

  The Nonlinear Pendulum
  http://www.pgccphy.net/ref/nonlin-pendulum.pdf
  by D. G. Simpson, 2010

- Useful background information about pi (and AGM), 1996:

  The Quest for Pi
  http://crd-legacy.lbl.gov/~dhbailey/dhbpapers/pi-quest.pdf
  by David H. Bailey, Jonathan M. Borwein, Peter B. Borwein and Simon Plouffe

- Some random web pages of interest:

  True period of a pendulum
  http://www.numericana.com/answer/physics.htm#pendulum

Simple gravity pendulum
http://www.scientificlib.com/en/Physics/LX/Pendulum.html

Arithmetic-geometric mean
http://en.wikipedia.org/wiki/Arithmetic%E2%80%93geometric_mean
http://en.wikipedia.org/wiki/AGM_method

- Links to my source code:

  Program to test agm (arithmetic-geometric mean) function
  http://leapsecond.com/tools/agm.c (agm.exe)

  Program to test 8 different pendulum period calculation methods
  http://leapsecond.com/tools/pend13.c (pend13.exe)

- A nice collection of links about pendulum period, especially issues of large amplitude, is found on this page. "Without the small-angle approximation the differential equation is not the simple harmonic motion equation and the period is not independent of the amplitude. The exact period can't be written in simple closed form, but it can be written as an elliptic integral of the first kind (which can be numerically approximated) or written as a power series expansion. Short of that, various other approximations are also available."

  Large Amplitude Pendulums
  http://www.snow.edu/larrys/PHYS2215/largeamplitudependulum.html
  by Larry Smith (Snow College), 2009+

- Copies of my HSN pendulum papers and links to programs and data:

  http://www.leapsecond.com/hsn2006/

/tvb