

Re: GPSMAP 60Cx or 76Cx?

Source: <http://sci.tech-archive.net/Archive/sci.geo.satellite-nav/2006-06/msg00196.html>

- *From:* Terje Mathisen <terje.mathisen@xxxxxxxxxxxxxx>
 - *Date:* Wed, 07 Jun 2006 22:01:16 +0200
-

Wolfgang S. Rupprecht wrote:

Terje Mathisen <terje.mathisen@xxxxxxxxxxxxxx> writes:

If you want to do trig "right", then you probably want to return exact (i.e. correctly rounded) results for all possible inputs: This requires you to do range reduction with a value for pi which is accurate to more than 1024 bits (which is the range of the exponent in IEEE double precision).

Thanks! That sounds more involved than I had realized. It also

Indeed, it is a lot more hairy than it seems at first glance.

nicely explains something that bugged me for a while --- why the BSD math library takes input in radians.

It is using radians because that is the natural base for all trig formulas:

I.e. for very small angles $\sin(\alpha) \approx \tan(\alpha) \approx \alpha$ which follows from the classic Taylor series for these functions:

$$\begin{aligned}\sin(a) &= a - a^3/3! + a^5/5! - a^7/7! \dots \\ \cos(a) &= 1 - a^2/2! + a^4/4! - a^6/6! \dots\end{aligned}$$

and

$$\tan(a) = \sin(a) / \cos(a)$$

It should be obvious that applying a scale factor to the input (i.e. $1/2 * \pi$) corresponds to a similar adjustment to each of the terms:

$x = a / s$ where s is the scale factor

$$\sin(x) = sx - x^3 * (s^3/3!) + x^5 * (s^5/5!) - \dots$$

Re: GPSMAP 60Cx or 76Cx?

The Cheby polynoms are very similar, but each term is adjusted slightly so as to minimize the maximum error over the entire input range given that the polynomial is truncated after a fixed number of terms.

Anyway, for maximum precision you still need to jump through some hoops to keep more intermediate bits around: Either you can use the "easy" solution taken by x86 cpus which is to work internally with 16 extra bits, i.e. extended precision has 16-bit sign+exponent and 64 bits of mantissa, or you have to fake it by splitting the input into two parts which together represents the actual value.

Anyway, as might be obvious by now, this has been sort of a hobby of mine since my high school days, and I've written several extended and/or arbitrary precision fp libraries.

I actually wrote a complete 128-bit library around new year 1994/95 so that I could verify the precision of the sw workaround we developed for the infamous Intel FDIV (and FPATAN) bug.

Send me email if you want the gory details, I'm probably far, far off-topic by now. :-)

Terje

Did you not see a significant term at half that period, i.e. 11:58?
This is the GPS sat period, so the constellation should repeat with this period, but over just a few weeks, the day/night difference will probably make one of the error peaks much more noticeable than the other?

No, for the same reason Dale pointed out. There are lots of similar excursions though. I assume this is because the SV's follow each other in a fairly tight track and if there is multipath from one SV there will be similar multipath from the next one to come by in 2-3 hrs. I wonder if it is possible to look at the times for the multipath and SV positions and use ray-tracing methods to map out where the multipath is generated.

-wolfgang

- <Terje.Mathisen@xxxxxxxxxxxxxx>
"almost all programming can be viewed as an exercise in caching"