

## Re: Appoximating large numbers

---

*Source:* <http://sci.tech-archive.net/Archive/sci.math.num-analysis/2006-06/msg00085.html>

---

- *From:* Roy Stogner <[roystgnrNO@xxxxxxxxxxxxxxxxxxxxxx](mailto:roystgnrNO@xxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Thu, 08 Jun 2006 15:41:14 GMT
- 

On Thu, 08 Jun 2006 08:18:52 -0400, Bit Twiddler wrote:

"Roy Stogner" <[roystgnrNO@xxxxxxxxxxxxxxxxxxxxxx](mailto:roystgnrNO@xxxxxxxxxxxxxxxxxxxxxx)> wrote in message  
[news:pan.2006.06.08.02.16.53.405175@xxxxxxxxxxxxxxxxxxxxxx](mailto:news:pan.2006.06.08.02.16.53.405175@xxxxxxxxxxxxxxxxxxxxxx)

Um, I didn't mean to hit a nerve or anything.

You didn't – if I came across as offended (or even excited), well, it's hard to convey emotional overtones correctly via text.

Why can't you use floating point numbers? A standard double precision floating point number will correctly represent the first 15 digits of any 300 digit number, and it'll do so in 64 bits instead of the 3.3 to 8 bits-per-digit that you're currently using. If your software package (and ideally your CPU) supports quadruple precision floating point, that'll give you the correct first 30 digits of any 16,000 digit number.

Knowing the first 30 digits of a number as long as 16K digits would definitely be a step in the right direction!

Thank you very much for your response Roy. I'd appreciate hearing about the 'obvious' idea that I am missing.

You won't find hardware support for quadruple precision everywhere, but it looks like you're using an interpreted instead of a compiled language, so I assume speed isn't an issue and you can use a software implementation. I don't know what exists for Ruby, but common phrases you can search for to try to find 128-bit floating point datatypes are "quadruple precision", "double double", and "long double".

If you can't find anything already existing that works for you, you'll need to code up your own scientific notation. I'm not familiar with Ruby

## Re: Appoximating large numbers

so I can't say what would work well – perhaps a `BigNum` for a mantissa and a `FixNum` for an exponent? If you've got so many big numbers that you're worried about space, however, you may need to use a different programming language entirely. Ruby seems to do on-the-fly conversions between fixed and variable length integers, and any language that does that is probably spending more than a few bytes per number just on overhead.

---

Roy Stogner

.