

Re: JSH: What's happening now?

Source: <http://sci.tech-archive.net/Archive/sci.math/2005-04/msg01978.html>

- *From:* "Tim Peters" <tim.one@xxxxxxxxxxxxx>
 - *Date:* Thu, 14 Apr 2005 00:49:14 -0400
-

....

[Proginoskes, to JSH]

> One more random note about your SF Theorem, as posted in your Google
> Group "Surrogate Factoring":

>> solving for y gives

>>

>> $y = (M^2 - Ax)/x^2$

>>

>> and solving for z using

>> [...]

>> and I can just solve to get

>>

>> $y = (Ax - M^2)/x^2$

> I suspect one of them is a typo.

How hostile of you.

> BTW, the closest that the SF Theorem can get to an algorithm is one
> where you're given M, then choose j, k₁, and f₁, to get a single
> factor (b₂) of M, which may or may not be an integer. (The values of
> T, Ax, Az, f₂, b₁, b₂, and k₂ are forced once the choices for M, j,
> k₁, and f₁ are made.)

Whew -- you've sure got more patience than I have! Can you pick one of the two expressions for `y` above such that the rest of the algebra actually looks right?

> Since k₁ and f₁ may need to be rational in order for b₂ to be an
> integer, this means you have to pick FIVE integers (j, the numerator
> and denominator of k₁, and the numerator and denominator of f₁) for
> each factoring attempt. This seems inefficient, because the "naive"
> algorithm picks ONE integer between 1 and M for each factoring attempt.

Ya, except that for a large two-prime product M=pq, where p is within a small factor of q, picking an int at random between 1 and M has a negligible chance of finding a non-trivial factor; while James keeps saying that his

Re: JSH: What's happening now?

"theorem" has a 50% chance of finding a non-trivial factor.

There's no justification for that I've seen. Nora Baron guessed that James thinks "OK, there are only 4 factors-- 1, p, q, and M --and half of those are non-trivial, so the odds are 1 in 2 of working". I guessed instead (although it's related) that he managed to confuse himself twice over with:

[JSH]

...

but if M has two prime factors p_1 and p_2 there are two cases out of that infinity for two factors that are rational:

$$f_1 = a/b, f_2 = (b p_1 p_2)/a$$

or

$$f_1 = (a p_1)/b, f_2 = (b p_2)/a$$

where a and b are non-zero integers coprime to $p_1 p_2$.

Confusion #1: "there are only two cases" (you explicitly showed that those two forms don't cover all the possibilities).

Confusion #2: "there are only two cases, and case #2 finds a non-trivial factor, so 1 out of 2 cases works, and 'the math' is equally likely to find either: 50% ". Of course that argument is wrong -- but I'm not sure he's actually making it.

If he had "a reason" for saying 50% other than those, I missed it -- each time I've seen the claim, it's pulled out of thin air, with no visible support of any kind; e.g.,

[JSH]

It's like, mathematically, it's a moot point, which indicates that it will factor in a way that appears to be truly random, giving you non-trivial factors about 50% of the time.

> Of course, it depends on how you pick your integers; it just seems
> wasteful.

That's my guess as to why he won't test it -- some part of him has to suspect it's going to do no better than picking gcd candidates at random. I should note that most previous algorithms in this family did worse than that: when you specify an algorithm for marching thru all the possibilities, it's not really random, and since there's no actual useful connection here between "rational factors" and integer factors, a poor-quality pseudo-random sequence is likely to do worse than picking candidates at random (a high-quality random sequence will eventually find a factor; a poor-quality sequence can systematically miss all multiples of p and q).

Re: JSH: What's happening now?

- **References:**

- ◆ **JSH: What's happening now?**

- ◇ From: jstevh

- ◆ **Re: JSH: What's happening now?**

- ◇ From: Proginoskes

- Prev by Date: **Re: Complete Solution ODE**

- Next by Date: **Re: combinatorial puzzle**

- Previous by thread: **Re: JSH: What's happening now?**

- Next by thread: **Re: JSH: What's happening now?**

- Index(es):

- ◆ **Date**

- ◆ **Thread**