Re: Bringing back an old tetration curiosity?

Source: http://sci.tech−archive.net/Archive/sci.math/2007−09/msg04416.html

- *From*: mike3 <mike4ty4@xxxxxxxxx>
- *Date*: Wed, 19 Sep 2007 23:32:35 −0700

On Sep 17, 3:31 pm, theronr...@xxxxxxxxx wrote:

> On Sep 17, 9:45 pm, mike3 <mike4...@xxxxxxxxx> wrote:
>
> > On Sep 17, 11:22 am, theronr...@xxxxxxxxx wrote:
> >
> > > On Sep 17, 6:12 pm, theronr...@xxxxxxxxx wrote:
> >
> > <snip>
> >
> > > After rethinking I think that the above statement is a bit
> > > speculative. However I'm sure that there is no continuous f:
> > > (−2; +oo)
> > > −> R that satisfies (1), (2) and (3). Here is why:
> > >
> > > http://img403.imageshack.us/img403/1466/sqrttetrxjx1.png
> > >
> > > This is a 5000x5000 plot of f(x) for −1 <= x <= 0 <= y <= 1.
> > >
> > > Theron
> >
> > This is interesting. There appears to be a gentle undulation
> > in it, oddly enough. It's intriguing to examine that in light
> > of a graph of ^x (0.1) on the integers, which oscillates quite
> > a bit.

> > I'd be curious to see your program, to root out various sources
> > of approximation error, for example, though.
>
> The branching doesn't look to me as a side effect of possible rounding
> errors. I can send you the C source code of the program, but I would
> like an independent verification of the result. If your program
> produces the same result (the same branching) then this will reaffirm
> the result. It is not difficult to write such a program.
>
> Theron

How did you write the program, exactly? That's why I'd like to see it.
My program just mirrors and recurses starting with an integer and
then plots all the iterates on the graph. This gives a more−or−less
sparse
graph, though. However on closer inspection it seems you might be
right −− there are points on the graph that make it seem sort of
"fuzzy" (in the way that would suggest the behavior you observed).
Decreasing precision of the numbers used did not appear to change
the graph. Of course I didn't change the _format_ of the number to
something (that might, say, handle overflows, for example,
differently) else, however.

This would suggest then that the tetrational function is _close_ to
symmetrical but not exactly so. I'd be curious to know if the
difference
between the value of ^0.5 sqrt(2) obtained using this method and that
obtained by assuming it is the square tetraroot of sqrt(2) is small
enough that perhaps the _correct_ curve for sqrt(2)'s tetrational
function would be able to pass through it at 0.5...

Regardless, the symmetry is obviously _close_, but no _cigar_.
I'd be curious to hear some more opinions from others here on
this group. It seems though like the guy's original idea, at least
in that form, is likely another dead end...

.