

# Re: FLTMA: A little group theory

---

*Source:* <http://sci.tech-archive.net/Archive/sci.math/2006-10/msg06644.html>

---

- *From:* "Chip Eastham" <[hardmath@xxxxxxxxxx](mailto:hardmath@xxxxxxxxxx)>
  - *Date:* 24 Oct 2006 09:51:28 -0700
- 

The Dougster wrote:

Chip Eastham wrote:

The Dougster wrote:

The Dougster wrote:

The Dougster wrote:

Chip Eastham wrote:

The  
Dougster  
wrote:

Ah.  
The  
order  
of  
 $-1$   
is  
 $2$ .  
 $|$   
 $<$   
 $-1$   
 $>$   
 $|$   
 $=$   
 $2$ .  
 $<$   
 $-1$   
 $>$   
 $=$   
 $\{$   
 $-1,$   
 $1$

Re: FLTMA: A little group theory

}.

How  
do  
we  
know  
that  
if  
 $w^n$   
 $\equiv$   
 $-1$   
mod  
 $z$   
that  
 $|$   
 $\langle$   
 $-1$   
 $\rangle$   
 $|$   
divides  
 $|$   
 $\langle$   
 $w$   
 $\rangle$   
 $|$   
?

Since  $\langle -1 \rangle$   
is a  
subgroup of  
 $\langle w \rangle$ , order  
of  $-1$  (two)  
divides  
the order of  
 $w$ .

Yipee! We're starting to use  
group theory to explore  
FLT!

<http://www.mathpages.com/home/kmath264.htm>

I think I see this more clearly today. If some power of  $w \equiv -1 \pmod{z}$   
then, knowing  $w^0 = 1$ , we have  $\{ 1, -1 \} \leq \langle w \rangle$  and so  
 $|\langle -1 \rangle|$  divides  
 $|\langle w \rangle|$ , where  $\leq$  means "is a subgroup of".

Re: FLTMA: A little group theory

I see in many sources on the web that without loss of generality, certain conclusions may be made from  $a^n + b^n = c^n$  in  $\mathbb{Z}$ . I have concluded, with help here in sci.math, that exactly one of  $\{x,y,z\}$  is even, and  $x < y < z < x+y$ . It might be more useful to give up  $x < y < z < x+y$  and find instead that, say,  $y$  is even, as some web sources have. I am still searching with Google for "Fermat's last theorem" and "without loss of generality" OR "elementary". I want to get that stuff out of the way, and certainly deduce as much as I can that might be useful later.

Nearly a month now with no tobacco during the day, when I am out.

An equation I have seen in the elementary results is  $x^p + y^p \equiv x+y \pmod p$ , or something similar. That would make 4 equations in 4 unknowns.

Doug

I think it would be interesting to make a targeted search for solutions with  $n = 3$ , the smallest possible prime, or even better, to develop a proof that no such solutions exist.

Using what we have already shown, we need coprime  $x,y,z$  such that:

$$\begin{aligned} (z/y) \pmod x &\text{ has order } 3 \\ (z/x) \pmod y &\text{ has order } 3 \\ (x/y) \pmod z &\text{ has order } 6 \text{ \& } (x/y)^3 \equiv -1 \pmod z \end{aligned}$$

where  $0 < x < y < z < x+y$  and  $xyz$  even.

Thus  $\phi(x)$  and  $\phi(y)$  must be divisible by 3, and  $\phi(z)$  must be divisible by 6. Thus  $z = 7, 9, 13, 14, 19, 18,$  etc. are candidates.

Chip

OK. I have done that in Mathcad, which doesn't share nicely in Usenet, and am working in VB4 now, so I can do something up which can be

Re: FLTMA: A little group theory

shared.

Is that  $xyz$  even or  $x+y+z$  even?

With  $x,y,z$  pairwise coprime, the notion that exactly one of  $x,y,z$  is even is expressed equally well by  $xyz$  even or  $x+y+z$  even, but I meant the former.

I wonder if I have access to a language in which operators can be overloaded to make the source code look more like abstract algebra and less like binary arithmetic. Hmm.

I like Prolog because one freely defines the relations/predicates that make up the language (and they are readily "overloaded" for different datatype, albeit at the expense of weak/runtime typing). However you may want to look at some "functional" languages like Haskell or ML. Lisp and descendants like Scheme are more "function" friendly, but I don't know how important infix notation is for you.

Can anyone in sci.math suggest an appropriate limit for such a search? Would 32-bit integers do all right? I think that's 16-bits of value before the table  $a*b$  in  $Z/nZ^*$  begins to overflow. There may be other limits. Shall we say, to  $z < 65535$ ?

Yes, I think this would be plenty interesting, to run  $z$  out to 16 bit unsigned limits, implemented in 32-bits for taking powers, etc.

regards, chip

.