

[> ### Example 2.3. p = 13 for (2.6).mws

A role for generalised Fermat numbers

by

John B. Cosgrave and Karl Dilcher

A reminder of the meaning of our paper's congruence (2.5): for $n = p^\alpha q_1^{\beta_1} q_2^{\beta_2} \dots q_s^{\beta_s}$, distinct primes $p, q_1, q_2, \dots, q_s = 1, -1, -1, \dots, -1 \pmod{6}$ we seek solutions of the Gauss factorial congruence

$$\text{floor}\left(\left(\frac{1}{6} \{n-1\}\right)_n!\right) = 1 \pmod{n}$$

In this Maple-to-pdf conversion we exhibit all solutions of our paper's congruence (2.6) for the prime $p = 13$, the second Jacobi prime.

It should be emphasised that for $p = 13$ the ' $s = 8$ ' is the current limit for which we can make a definitive statement concerning the exact number of solutions of (2.6). A similar statement for $s = 9$ would require knowing the complete factorisation of the 286-digit $13^{2^8} + 1$ (that '8' being $9 - 1$)

Note. At $p = 13$ we have $\text{ord}_p\left(\frac{1}{3} \{p-1\}\right)! = 3 \cdot 2^2$. and $\text{ord}_p\left(\frac{1}{6} \{p-1\}\right)! = 3 \cdot 2^2$.

This non-standard Jacobi prime is truly remarkable: it is the only 1-exceptional prime (at $M = 3, 6$) to 10^{14} , and thus is the only prime (necessarily Jacobi) for which the congruences (2.5) or (2.6) of our "*A role for generalized Fermat numbers*" paper can have solutions with $\alpha = 2$. Indeed, because 13 is not 2-exceptional (and thus cannot have any higher level of exceptionality) then congruences (2.5) or (2.6) of our "*A role ...*" paper cannot have any solutions with $p = 13$ for $\alpha > 2$.

Procedures

```
> with(numtheory): ### needed for 'order'
  with(combinat): ### needed for 'choose'
  ### 01:
  PI := proc(n, M, i) local k, r; r := 1:
    for k from floor(((i-1)*(n-1)/M + 1)) to i*(n-1)/M do
      if igcd(n, k) = 1 then r := mods(r*k, n); fi; od; r; end:
  ### 02:
  PRFAC := proc(le, la, p, alpha) local r, k, MOD; r := le; MOD := p^alpha;
    for k from (le+1) to la do r := mods(r*k, MOD) od; r; end:
  ### 03:
  the_ones := proc(n, M) local L, p; L := []: for p in factorset(n) do if p mod M = 1 then
    L := [op(L), p] fi od; L; end:
```

```

#### 04:

the_minus_ones := proc(n, M) local L, p; L := []: for p in factorset(n) do if mods(p, M) = -1
then L := [op(L), p] fi od; L; end;

#### 05:

Pow := proc(n, p) local t, a; t := n; a := 0: while t mod p = 0 do t := t/p; a := a+1: od: a;
end;

#### 06:

#### "more" simply means that 's' (in application) > 1
#### (i.e., there is 'more' than ONE 'q')

#### Bear in mind that in using this procedure we are making the understanding that p
#### occurs only to the 1st power, and 'w' will be square-free ('primitive' solutions)

#### 'L' REPLACES the INITIAL 'w'
#### 'w' becomes a local variable, DEFINED as the product of the members of L (the 'q')

PI6_more_modified := proc(L, s, p, fac) local w, EF, FAC, R;

if s < 2 then lprint(`Remember that s should be greater than 1.`); RETURN(); fi;

w := mul(q, q in L);

EF := mods(w&^( (p - 1)/3 ), p):    ### Euler-Fermat element for s > 1.

FAC := mods( fac&^(2^s), p):        ### This is the Gauss 6 closed form mod p

if mods(w, 6) = 1 and s mod 2 = 0 then

R := mods(EF * FAC, p);

elif mods(w, 6) = 1 and s mod 2 = 1 then

R := mods(FAC / EF, p);

elif mods(w, 6) = -1 and s mod 2 = 0 then

R := mods(1 / ( EF * FAC ), p);

elif mods(w, 6) = -1 and s mod 2 = 1 then

R := mods(EF / FAC, p);

fi;

R; end;

#####
#### The following PI6 - of which we make only limited use - tests some individual n-values
#### It allows for having 'alpha' > 1

#### 07:

#### The following PHI6 uses the D.H. Lehmer formulae for the
#### PHI-values of w = 1/-1 (mod 6) for w having NO prime factor = 1 (mod 6)

PHI6 := proc(w) local s; s := nops(factorset(w)):

if w mod 6 = 1 then (phi(w) + 2^(s+1))/6 elif mods(w, 6) = -1 then (phi(w) - 2^(s+1))/6 fi;
end;

#### 08:

PI6 := proc(n) local p, a, Gf, w, s, signs, PHIw6,
Sw, Q, PARI, EF1, q1, sign1, EF, Rpa, Rw, R;

p := op(the_ones(n, 6)):                                ### This gives 'p'
a := Pow(n, p):                                         ### This gives 'a', i.e. 'alpha'

if a = 1 then Gf := PRFAC(1, (p-1)/6, p, 1) elif a > 1 then Gf := PI(p^a, 6, 1) fi;

### This is ((p^a - 1)/6)_p! mod p^a, speeded up in the case a(alpha) = 1

w := n/(p^a):                                         ### This is 'w'
PARI := proc(w) if mods(w, 6) = 1 then 1      ### This is '1' if w = 1 (mod 6)
elif mods(w, 6) = -1 then -1 fi end:   ### but is '-1' if w = -1 (mod 6)

s := nops(factorset(w)):    ### This is 's'
signs := (-1)^s:          ### This is '-1' at ODD 's', and '1' at EVEN 's'

```

```

PHIw6 := PHI6(w):

Sw := factorset(w);           ### The set of all ('s' of) the 'q'
Q := mul(q, q = Sw);         ### The product of all ('s' of) the 'q'

### We need TWO EULER-FERMATS (EF1 and EF) to distinguish between s = 1 and s > 1:

if s = 1 then

    q1 := op(1, factorset(w));           ### This is 'q[1]'
    sign1 := (-1)^((p+q1)/6);          ### the sign element in the closed form
    EF1 := mods(q1&^(phi(p^a)/6), p^a);  ### the Euler-Fermat element for q[1].
                                                ### NOTE the 6th-root
    Rpa := mods(sign1 * EF1 * Gf^2, p^a):
    Rw := mods((-1)^((p-1)/6)/p&^PHIw6, w): ### Note the EXTRA SIGN element here at Gauss
6

    R := mods(chrem([Rpa^PARI(w), Rw], [p^a, w]), n):
                                                ### Note the 1/Rpa, as with Gauss 4 closed forms

elif s > 1 then

    EF := mods(Q&^(phi(p^a)/3), p^a):   ### Euler-Fermat element for s > 1. Note the
3rd-root                                         signs
    Rpa := mods(EF^signs*Gf^(2^s), p^a):  ### Note the SIGN element in the EF term
    Rw := mods(1/p&^PHIw6, w):            ### There is NO SIGN element here at s > 1

    R := mods(chrem([Rpa^PARI(w), Rw], [p^a, w]), n):

fi; R; end:

```

The Gauss 6 support primes for Jacobi $p = 13$ (completely factored to $13^{27} + 1$)

```

p := 13:

level||2[p] := [5, 17]:           ### These divide  $(13 - 1) * (13 + 1) * (13^2 + 1)$ 
level||3[p] := []:                ### No 'q' divides  $13^{(2^2)} + 1$ 
level||4[p] := []:                ### No 'q' divides  $13^{(2^3)} + 1$ 

level||5[p] := [2657, 441281]:    ### These divide  $13^{(2^4)} + 1$ 
level||6[p] := [1601, 10433]:     ### These divide  $13^{(2^5)} + 1$ 
level||7[p] := [257, 36713826768408543617]:  ### These divide  $13^{(2^6)} + 1$ 
level||8[p] := []:                ### Have a COMPLETE factorisation for  $13^{(2^7)} + 1$ 
1
                                                ### but NO 'q' = -1 (mod 6)
                                                ### a factordb factorisation

### support COMPLETE to this level

LEV := 8:

LEVEL||LEV[p] := []:
for lev from 2 to LEV do
for q||lev in level||lev[p] do LEVEL||LEV[p] := [ op(LEVEL||LEV[p]), q||lev ]; od od:
print(` `); LEVEL||LEV[p];

```

```
> seq(isprime(q), q = LEVEL||LEV[p]);
      true, true, true, true, true, true, true, true, true
```

(2.2)

```
> seq(q mod 6, q = LEVEL||LEV[p]);      5, 5, 5, 5, 5, 5, 5, 5
```

(2.3)

```
> print(``);
[[2], seq((p - 1)*(p + 1)*(p^2 + 1) mod q, q = level||2[p]));
for LEV from 3 to 8 do if level||LEV[p] <> [] then
print([[LEV], seq(p&^(2^(LEV - 1)) + 1 mod q, q = level||LEV[p])]); fi od;
[[2], 0, 0]
[[5], 0, 0]
[[6], 0, 0]
[[7], 0, 0]
```

(2.4)

> ### The following determines any q-support for which q^2 is a factor:

```
> print(``);
[[2], seq((p - 1)*(p + 1)*(p^2 + 1) mod q^2, q = level||2[p]);
for LEV from 3 to 8 do if level||LEV[p] <> [] then
print([[LEV], seq(p&^(2^(LEV - 1)) + 1 mod q^2, q = level||LEV[p])]); fi od;
[[2], 10, 238]
[[5], 1676567, 40023745419]
[[6], 2230193, 15190448]
[[7], 12593, 604231368639226848823138615882197189674]
```

(2.5)


```
> print(``;
for q in level||2[p] do if (p - 1)*(p + 1)*(p^2 + 1) mod q^2 = 0 then print([2], q) fi od;
for LEV from 3 to 8 do if level||LEV[p] <> [] then
for q in level||LEV[p] do if p&^(2^(LEV - 1)) + 1 mod q^2 = 0 then print([LEV], q) fi od: fi od;
```

(2.6)

Thus, in the case of $p = 13$, there are no square supports.

```
>          p := 13;
fac := PRFAC(1, (p-1)/6, p, 1);
ord := ifactor(order(fac, p));
          p:=13
          fac:=2
          ord:=(2)^2 (3)
```

(1)

s = 2. The long-known solution ($1105 = 13 \cdot 5 \cdot 11$), and then one extra using the factor 13^2 (also long-known)

```
>          p := 13;
fac := 2: ### THE ABOVE PRE-COMPUTED VALUE OF fac
LEV := 2:
LEVEL||LEV[p] := []:
for lev from 2 to LEV do
```

```

for q||lev in level||lev[p] do LEVEL||LEV[p] := [ op(LEVEL||LEV[p]), q||lev ]; od od:
print(``); S_potential||p := LEVEL||LEV[p];
count := 0;
SOLN_L := []:
print(``); print(______); print(``);
for L_ in choose(S_potential||p, LEV) do if PI6_more_modified(L_, LEV, p, fac) = 1 then
    count := count+1;
print(``); print(L_); print(``); lprint(`Here is a solution:``); print(``);
print(p*mul(j, j = L_));
    SOLN_L := [op(SOLN_L), p*mul(j, j = L_)];
print(``); lprint(`which has`, length(p*mul(j, j = L_)), `digits.`); print(______);
fi; od:
print(``); lprint(`There were`, count, `solutions altogether.``); print(______);
print(``);

```

S_potential13 := [5, 17]

[5, 17]

`Here is a solution:`

1105

`which has`, 4, `digits.`

`There were`, 1, `solutions altogether.`

(3.1)

> print(``); ifactor(1105);

(5) (13) (17)

(3.2)

> print(``); n := 13*1105; ifactor(n);
[PI6(n)]; ### Verifying the 13^2 element

n := 14365
(5) (13)^2 (17)
[1]

(3.3)

There cannot be any solutions at $s=3, 4$ or 5 , because of paucity of support primes.

Nor is there a solution at the singleton $s=6$.

Then:

► $s=7$. Produced 4 solutions (with 22, 36, 38 and 41 digits), and then four extra solutions using the factor

Those extra four solutions verified using the general PI6 procedure.

```
> p := 13:
fac := 2: ### THE ABOVE PRE-COMPUTED VALUE OF fac
LEV := 7:
LEVEL||LEV[p] := []:
for lev from 2 to LEV do
for q||lev in level||lev[p] do LEVEL||LEV[p] := [ op(LEVEL||LEV[p]), q||lev ]; od od:
print(``); S_potential||p := LEVEL||LEV[p];
count := 0:
SOLN_L := []:
print(``); print(______); print(``);
for L_ in choose(S_potential||p, LEV) do if PI6_more_modified(L_, LEV, p, fac) = 1 then
    count := count+1:
print(``); print(L_); print(``); lprint(`Here is a solution:``); print(``);
print(p*mul(j, j = L_));
SOLN_L := [op(SOLN_L), p*mul(j, j = L_)];
print(``); lprint(`which has`, length(p*mul(j, j = L_)), `digits.`); print(______);
fi; od:
print(``); lprint(`There were`, count, `solutions altogether.``); print(______);
print(``);
```

S_potential13:=[5, 17, 257, 441281, 1601, 10433, 257, 36713826768408543617]

[5, 17, 257, 1601, 2657, 10433, 441281]

`Here is a solution:``

5561638076329536617585

`which has`, 22, `digits.`

[5, 17, 257, 1601, 2657, 10433, 36713826768408543617]

`Here is a solution:``

462718804759206810163766270220023345

`which has`, 36, `digits.`

[5, 17, 257, 1601, 2657, 441281, 36713826768408543617]

`Here is a solution:``

19571457575284917127947564793344399665

`which has`, 38, `digits.`

```
[5, 257, 1601, 2657, 10433, 441281, 36713826768408543617]
```

```
`Here is a solution: `
```

```
12011118640173384729169231969938948335585
```

```
`which has` , 41, `digits. `
```

```
`There were` , 4, `solutions altogether. `
```

(4.1)

```
> for n in SOLN_L do print(``); print(ifactor(13*n), PI6(13*n)); od;
```

```
(5) (13)^2 (17) (257) (1601) (441281) (10433) (2657), 1
```

```
(5) (13)^2 (17) (257) (1601) (36713826768408543617) (10433) (2657), 1
```

```
(5) (13)^2 (17) (257) (1601) (36713826768408543617) (441281) (2657), 1
```

```
(5) (13)^2 (257) (1601) (36713826768408543617) (441281) (10433) (2657), 1
```

(4.2)

s = 8. No solution at this singleton

```
> p := 13:
fac := 2: ### THE ABOVE PRE-COMPUTED VALUE OF fac
LEV := 8:
LEVEL||LEV[p] := []:
for lev from 2 to LEV do
for q||lev in level||lev[p] do LEVEL||LEV[p] := [ op(LEVEL||LEV[p]), q||lev ]; od od:
print(``); S_potential||p := LEVEL||LEV[p];
count := 0:
SOLN_L := []:
print(``); print(______); print(``);
for L_ in choose(S_potential||p, LEV) do if PI6_more_modified(L_, LEV, p, fac) = 1 then
    count := count+1:
print(``); print(L_); print(``); lprint(`Here is a solution:`); print(``);
print(p*mul(j, j = L_));
SOLN_L := [op(SOLN_L), p*mul(j, j = L_)];
print(``); lprint(`which has` , length(p*mul(j, j = L_)), `digits.`); print(______);
fi; od:
print(``); lprint(`There were` , count, `solutions altogether. `); print(______);
print(``);
```

```
S_potential13:=[5, 17, 2657, 441281, 1601, 10433, 257, 36713826768408543617]
```

There were, 0, `solutions altogether.'

(5.1)

- COMPLETE to here (as there are no square supports), level 8 (remember there is no level 8 support)

An ideal would be the (possibly unrealistic) factorisation of the 286-digit $13^{2^8} + 1$ (needed for Gauss 6, level 9)