

Fixed points and Two-cycles of the discrete logarithm

Joshua Holden
Rose-Hulman Institute of Technology

Question (Brizolis) Given a prime $p > 3$, is there always a pair (g, h) such that g is a generator modulo p , $1 \leq h \leq p - 1$, and

$$g^h \equiv h \pmod{p} ? \quad (1)$$

- I.e., is there always a generator g such that \log_g has a fixed point?
- What is the asymptotic proportion of pairs (g, h) ?
- Does it matter if g is a generator? What about h ?
- What about k -cycles instead of fixed points?
- How much does the induced graph look like a “random graph”?
- Applications to pseudorandom generators.

Heuristics for fixed points

- Notation:
 x RP means $\gcd(x, p - 1) = 1$
 x PR means x is a generator mod p
 x RPPR means x is both RP and PR
 x ANY means no conditions on x
- $N_{(1),g \text{ ANY},h \text{ RP}}(p) = \phi(p - 1)$

Theorem (Zhang, independently by others)

$$N_{(1),g \text{ PR},h \text{ RPPR}}(p) \approx \phi(p - 1)^2 / (p - 1)$$

- Asymptotically answers Brizolis
- Campbell and Pomerance: complete answer soon
- Other cases require heuristic conjectures.

Fixed points

predictions: formulas

$g \setminus h$	ANY	PR	RP	RPPR
ANY	$\approx (p-1)$	$\approx \frac{\phi(p-1)^2}{(p-1)}$	$= \phi(p-1)$	$\approx \frac{\phi(p-1)^2}{(p-1)}$
PR	$\approx \phi(p-1)$	$\approx \frac{\phi(p-1)^2}{(p-1)}$	$\approx \frac{\phi(p-1)^2}{(p-1)}$	$\approx \frac{\phi(p-1)^2}{(p-1)}$
RP	$\approx \phi(p-1)$	$\approx \frac{\phi(p-1)^3}{(p-1)^2}$	$\approx \frac{\phi(p-1)^2}{(p-1)}$	$\approx \frac{\phi(p-1)^3}{(p-1)^2}$
RPPR	$\approx \frac{\phi(p-1)^2}{(p-1)}$	$\approx \frac{\phi(p-1)^3}{(p-1)^2}$	$\approx \frac{\phi(p-1)^3}{(p-1)^2}$	$\approx \frac{\phi(p-1)^3}{(p-1)^2}$

predictions: prime = 100057

$g \setminus h$	ANY	PR	RP	RPPR
ANY	100056	9139.46	30240	9139.46
PR	30240	9139.46	9139.46	9139.46
RP	30240	2762.23	9139.46	2762.23
RPPR	9139.46	2762.23	2762.23	2762.23

observed: prime = 100057

$g \setminus h$	ANY	PR	RP	RPPR
ANY	98506	9192	30240	9192
PR	29630	9192	9192	9192
RP	29774	2784	9037	2784
RPPR	9085	2784	2784	2784

Question (Two-cycles) Given a prime $p > 3$, how many pairs (g, h) are there such that such that there is some a between 1 and $p - 1$ such that

$$g^h \equiv a \pmod{p} \quad \text{and} \quad g^a \equiv h \pmod{p} ? \quad (2)$$

- Equivalently, $g^{g^h \pmod{p}} \equiv h \pmod{p}$.

- Consider “collision equation”

$$h^h \equiv a^a \pmod{p} . \quad (3)$$

- If (g, h) is a solution to (2) then $(h, a) \equiv (h, g^h)$ is a solution to (3).

- Sometimes but not always one to one.

- Two-cycles which are fixed points (“trivial”) correspond to $h = a$.

Heuristics for collisions

- (3) is much easier to compute: $O(p)$ exponentiations, not $O(p^2)$.
- Tally up the results by h^h in a table of all possible values.
- Want to keep this all in one memory space!
- Heuristic conjectures for (3) can be made rigorous?

collisions

predictions: formulas for **nontrivial** part

$a \setminus h$	ANY	PR	RP	RPPR
ANY	$\approx \sum \frac{ S_m ^2}{ T_m }$	$\approx \phi(p-1)$	$\approx \phi(p-1)$	$\approx \frac{\phi(p-1)^3}{(p-1)^2}$
PR	$\approx \phi(p-1)$	$\approx \frac{\phi(p-1)^2}{(p-1)}$	$\approx \frac{\phi(p-1)^2}{(p-1)}$	$\approx \frac{\phi(p-1)^3}{(p-1)^2}$
RP	$\approx \phi(p-1)$	$\approx \frac{\phi(p-1)^2}{(p-1)}$	$\approx \frac{\phi(p-1)^2}{(p-1)}$	$\approx \frac{\phi(p-1)^3}{(p-1)^2}$
RPPR	$\approx \frac{\phi(p-1)^3}{(p-1)^2}$	$\approx \frac{\phi(p-1)^3}{(p-1)^2}$	$\approx \frac{\phi(p-1)^3}{(p-1)^2}$	$\approx \frac{\phi(p-1)^3}{(p-1)^2}$

predictions (all): prime = 100057

$a \setminus h$	ANY	PR	RP	RPPR
ANY	290878.0	60480	60480	11901.7
PR	60480	39379.5	18278.9	11901.7
RP	60480	18278.9	39379.5	11901.7
RPPR	11901.7	11901.7	11901.7	11901.7

observed (all): prime = 100057

$a \setminus h$	ANY	PR	RP	RPPR
ANY	290582	60466	60531	12012
PR	60466	39490	18423	12012
RP	60531	18423	39326	12012
RPPR	12012	12012	12012	12012

Heuristics for two-cycles

- If $d = \gcd(a, h, p - 1) = 1$ then use collisions.

- If not, use

$$h^{h/d} \equiv a^{a/d} \pmod{p}. \quad (4)$$

- Similar arguments but require stronger heuristic conjectures.

Two-cycles

predictions: formulas for **nontrivial** part

$g \setminus h$	ANY – ?	PR	RP	RPPR
ANY	$\approx (p-1)$	$\approx \phi(p-1)$	$\approx \frac{\phi(p-1)^2}{(p-1)}$	$\approx \frac{\phi(p-1)^3}{(p-1)^2}$
PR	$\approx \phi(p-1)$	$\approx \frac{\phi(p-1)^2}{(p-1)}$	$\approx \frac{\phi(p-1)^2}{(p-1)}$	$\approx \frac{\phi(p-1)^3}{(p-1)^2}$
RP	$\approx \phi(p-1)$	$\approx \frac{\phi(p-1)^2}{(p-1)}$	$\approx \frac{\phi(p-1)^3}{(p-1)^2}$	$\approx \frac{\phi(p-1)^4}{(p-1)^3}$
RPPR	$\approx \frac{\phi(p-1)^2}{(p-1)}$	$\approx \frac{\phi(p-1)^3}{(p-1)^2}$	$\approx \frac{\phi(p-1)^3}{(p-1)^2}$	$\approx \frac{\phi(p-1)^4}{(p-1)^3}$

predictions (all): prime = 100057

$g \setminus h$	ANY	PR	RP	RPPR
ANY	200112	18278.9	60480	11901.7
PR	60480	18278.9	18278.9	11901.7
RP	60480	5524.5	18278.9	3597.1
RPPR	18278.9	5524.5	5524.5	3597.1

observed (all): prime = 100057

$g \setminus h$	ANY	PR	RP	RPPR
ANY	199366	18423	60531	12012
PR	60480	18423	18423	12012
RP	60142	5666	18277	3700
RPPR	18461	5666	5666	3700

Data collection

- Beowulf cluster, 19 nodes, 2 Pentium III processors each at 1Ghz
- C, MPI, OpenMP, OpenSSL (large integer calculations)
- 68 hours for five primes starting at 100000, all three tables

Algorithm overview

```
1: initialize
2: for  $j = 1$  to  $nump$  do
3:   find a new prime  $p$  and generator  $gen$ 
4:   for  $x = 1$  to  $p - 1$  do {distribute among
   nodes}
5:      $g = gen^x \bmod p$ 
6:     for  $y = 1$  to  $p - 1$  do {distribute
   among threads}
7:        $h = gen^y \bmod p$ 
8:       if  $x = 1$  then
9:          $b = h^h \bmod p$ 
10:        record  $b$ 
11:         $a = g^h \bmod p$ 
12:         $b = a^h \bmod p$ 
13:        if  $a = h$  then
14:          increase fixed point tally
15:          if  $b = h$  then
16:            increase two-cycle tally
17: tally collision data
18: correlate data
19: report data
```

C program (edited)

```
1 #pragma omp threadprivate
2   ( /* COPYIN VARS ... */ )
3
4 main(int argc, char *argv [])
5 {
6   /* VAR DEFINITIONS ... */
7
8   /* Init MPI */
9   MPI_Init(&argc, &argv);
10  MPI_Comm_size(MPI_COMM_WORLD,
11                &numnodes);
12  MPI_Comm_rank(MPI_COMM_WORLD,
13                &myid);
14
15  if ( argc >= 4 )
16    numthreads=atoi(argv[3]);
17
18  omp_set_num_threads(numthreads);
19
20 #pragma omp parallel
```

```

21  default ( private )
22  shared ( /* SHARED VARS ... */ )
23  copyin ( /* COPYIN VARS ... */ )
24  {
25  #pragma omp master
26  if ( myid == 0 )
27      /* INFORMATIVE MESSAGE ... */
28
29  /* INIT VARS ... */
30
31  for ( j=1; j<=nump; j++ )
32  {
33      nextprime(p);
34      primroot(gen, p);
35
36  #pragma omp barrier
37  #pragma omp master
38  {
39      /* INIT SHARED VARS ... */
40      hatab = (int (*)[2][2])
41          calloc(4*p, sizeof(int));

```

```
42     } /* end master */
43     #pragma omp barrier
44     #pragma omp flush
45
46     /* dist among nodes */
47     for (x = myid + 1; x < p;
48         x = x + numnodes)
49     {
50         mod_exp(g, gen, x, p);
51         gcd(d, g, p-1);
52         gisrp = is_one(d);
53         gcd(d, x, p-1);
54         gispr = is_one(d);
55
56         /* dist among threads */
57         #pragma omp for
58         for (y=1; y < p; y++)
59         {
60             mod_exp(h, gen, y, p);
61             gcd(d, h, p-1);
62             hisrp = is_one(d);
```

```
63     gcd(d, y, p-1);
64     hispr = is_one(d);
65
66     /* collision data */
67     /* only need first */
68     /* time through loop */
69     /* use node 0 */
70     if ( x == 1 )
71     {
72         mod_exp(b, h, h, p);
73         hatab[ /*...*/ ]++;
74     } /* end if */
75
76     mod_exp(a, g, h, p);
77     mod_exp(b, g, a, p);
78     if ( a == h )
79     {
80         fxcount[ /*...*/ ]++;
81         #pragma omp flush
82     }
83
```

```
84         if ( b == h )
85         {
86             tccount [ /*...*/ ]++;
87             #pragma omp flush
88         }
89
90         /* implicit barrier */
91     } /* end y loop */
92 } /* end x loop */
93
94 /* tally collision data */
95 /* same node as above */
96 /* only one thread (?) */
97 #pragma omp flush
98 #pragma omp barrier
99 #pragma omp single
100     if ( myid == 0 )
101         /* RUNNING TOTAL ... */
102     } /* end single */
103 #pragma omp flush
104
```

```
105     /* gather the data */
106     /* master for each node */
107     /* collect at root node */
108     #pragma omp barrier
109     #pragma omp master
110     {
111         for ( /* ... */ )
112         {
113             MPI_Reduce(
114                 &(fxcount [ /* ... */ ]),
115                 &(fxglob [ /* ... */ ]),
116                 1, MPI_INT, MPI_SUM,
117                 0, MPI_COMM_WORLD);
118             MPI_Reduce(
119                 &(tccount [ /* ... */ ]),
120                 &(tcglob [ /* ... */ ]),
121                 1, MPI_INT, MPI_SUM,
122                 0, MPI_COMM_WORLD);
123             haglob [ /* ... */ ] =
124                 hacount [ /* ... */ ];
125         }
```

```
126     } /* end master */
127     #pragma omp flush
128
129     /* report data */
130     #pragma omp master
131     if (myid == 0)
132     {
133         /* REPORT DATA ... */
134         fflush(stdout);
135     } /* end master thread/node */
136
137     #pragma omp barrier
138
139     #pragma omp master
140     free(hatab);
141
142     } /* end j loop */
143 } /* end parallel structure */
144
145 MPI_Finalize();
146
147 } /* end main */
```

Future work

- More data?
- Rigorous versions of these estimates.
- Average behavior questions.
- More Brizolis-type existence questions.