

## Chapter 2

# Warming Up: Integers, Sequences, and Experimental Mathematics

Motivational: quick overview, exciting trivia, second looks, startups ....

### 2.1 From the Lebombo Bone to OEIS

The oldest known record of a mathematical object is the fossilized Lebombo bone [more than 43,000 years old according to rigorous carbon dating (d’Errico et al. 1987)], displaying 29 tally marks. Just for fun: this means that the first integer ever recorded in human history happens to be a prime number! One could indeed say that the natural numbers 1, 2, 3, ... have fascinated humanity from the dawn of time, thus becoming engrained in human consciousness (Greathouse; Dehaene 1999).

Integer sequences are mathematical objects with a long history. Babylonian mathematics is a culture displaying interesting computationally oriented features, some involving sequences, as exemplified by the sequence of squares of numbers up to 59 and the sequence of cubes of numbers up to 32 appearing in clay tablets from Senkerah dating from 2000 B.C. (Allen 2016).

An interesting integer sequence is a pattern waiting to be discovered (and hence an interesting problem to be solved) and an evolving phenomenon to be witnessed, which appears to have a life of its own. It elicits passion and interest from the inquisitive, creative mind, opening it to an experiential approach in mathematics. And for undergraduate students, such sequences, in the author’s view, constitute one of the most powerful tools for advancing a high-impact learning in many mathematical areas, especially in number theory and its applications.

Arguably the most important resource on integer sequences today, luckily available for free, is the On-Line Encyclopedia of Integer Sequences (OEIS), <http://oeis.org>. Initiated by Neil J.A. Sloane in early 1964 (OEIS; NJA Sloane 2013) and continuously maintained and expanded by a group of enthusiasts and contributors, OEIS is a huge list of interesting sequences. Every one of them is identified by an OEIS code. For example, just to mention four classical examples, the sequence of

natural numbers (1, 2, 3, 4, 5, ...) is listed as A000027, the sequence of primes (2, 3, 5, 7, 11, ...) goes under the label A000040, while the sequence of Fibonacci numbers (0, 1, 1, 2, 3, 5, 8, ..., i.e., every term is the sum of the previous two, starting from the “seed” 0, 1) goes under the OEIS code A000045. Every listing is accompanied by a rich and useful set of related comments, references to published work, notes, links, theorems, available explicit formulas, program lines (MAPLE, Mathematica, etc.) generating the sequence, and a list of OEIS codes of closely related sequences. OEIS is a valuable research resource for the mathematical community including undergraduate students aspiring to perform meaningful work in, or with, mathematics.

## 2.2 Experimental Mathematics

OEIS lists, with detailed references, an amazing number of new sequences at the forefront of current research. N.J.A. Sloane is right in calling it “an endless source of open problems.” This definitely makes OEIS an important player and a useful collaborative tool in the emerging area known as experimental mathematics.

On the University of Copenhagen web page dedicated to “Experimental Mathematics in Number Theory, Operator Algebras, and Topology” (ExpMathDK 2016) can be found the following concise definition:

Experimental mathematics is the modus operandi of using computers for generating insight into pure mathematics.

Kurt Gödel, the mathematical visionary, anticipated—in an unpublished 1951 essay—a deep connection between the approaches taken in physics and experimental mathematics, with an interesting philosophical overtone [in the platonic view, mathematical truth is objectively “out there” to be discovered by the working mathematician (Gödel 1951)]:

If mathematics describes an objective world just like physics, there is no reason why inductive methods should not be applied in mathematics just the same as in physics.

We can say that from an experimental mathematics viewpoint, we learn to experiment with numbers (say integer sequences) and see how they behave. The hope is that interesting patterns and phenomena will appear along the way. Especially when more computing power is involved, thus making possible the manipulation of larger integers, it is easier to accept (with an eye to the above-mentioned Gödelian analogy with physics) that new emergent “mathematical phenomena” will pop up, pretty much like when new elementary particles and related phenomena arguably appear as a result of higher-energy collisions in more powerful particle accelerators.

Of course, computer experimentation is rewarding in itself as a potentially limitless exploration of the (mathematical) space. Far from being a blind walk, experimentation/computation in mathematics is rather a process of meaningful

intentionality, by creatively engaging the human person, with the hope that amazing facts will present themselves. This is, in fact, the destiny of the working mathematician, so eloquently and artfully presented by Frenkel (2013):

Engaging implies hard work. More often than not, at the end of the day (or a month, or a year), you realize that your initial idea was wrong, and you have to try something else. These are the moments of frustration and despair. You feel that you have wasted an enormous amount of time, with nothing to show for it. This is hard to stomach. But you can never give up. You go back to the drawing board, you analyze more data, you learn from your previous mistakes, you try to come up with a better idea. And every once in a while, suddenly, your idea starts to work. It's as if you had spent a fruitless day surfing, when you finally catch a wave: you try to hold on to it and ride it for as long as possible. At moments like this, you have to free your imagination and let the wave take you as far as it can. Even if the idea sounds totally crazy at first.

The implicit expectation is that the patterns and structures that appear in this experimental/computational process of discovery

- will receive proofs (or refutations for that matter) somewhere down the line (one may try partial proofs first), and
- will inspire in real time further meaningful experiments, adding to the wonder and the complexities of the problem.

Last but not least, if an interesting “artistic” pattern emerges in the process, it would be helpful to dwell on it further, since there is a good reason, waiting to be discovered, behind the observed symmetry. Also, you may want to hold on to such discoveries: they are good advertising for mathematics.

## 2.3 Primes

With an experimental mathematics mindset, let's turn back to A000040 (the sequence of primes, i.e., positive integers greater than 1 without any divisor other than 1 and themselves):

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, . . .

This is a tremendously important sequence for mathematics as a whole. Without an immediately discernible pattern, they form a rich data set that has fascinated mathematicians for millennia. In reference to the sequence of primes, Tim Gowers said (2002):

Although the prime numbers are rigidly determined, they somehow feel like experimental data.

Crucial results on primes go back to the Greek mathematician Euclid's *Elements* (2016) (about 300 BC). Among them:

- There are infinitely many primes.
- Every positive integer has a unique prime factorization (which makes prime numbers the multiplicative building blocks for the set of natural numbers).
- The importance of primes of a special form: thus, the number  $2^{n-1}(2^n - 1)$  is perfect (equals the sum of its proper divisors) whenever  $2^n - 1$  is prime.

Fast-forwarding to today, arguably the single most important mathematical object that monolithically encodes the sequence of primes (and is the decisive factor in the study of the distribution of primes) is Riemann's zeta function (Edwards 1974):

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s} = \prod_{p \text{ prime}} \frac{1}{1 - \frac{1}{p^s}}.$$

The zeta function can be analytically continued to a complex function that is holomorphic at every point except  $s = 1$  (where it has a simple pole). The seminal 1896 result known as the prime number theorem, proved independently by Jacques Hadamard and Charles Jean de la Vallée-Poussin through extending Riemann's ideas, gives an asymptotic form of the “prime counting function”  $\pi(x) = \#\{p | p \text{ prime}, p \leq x\}$ :

$$\lim_{x \rightarrow \infty} \frac{\pi(x)}{x / \ln x} = 1.$$

Some consequences of  $\pi(x) \approx x / \ln x$ :

- The asymptotic formula for the  $n$ th prime:  $p_n \approx n \ln n$ . This can be refined by the inequality  $n(\ln n + \ln \ln n - 1) < p_n < n(\ln n + \ln \ln n)$ , holding for  $n \geq 6$  (Rosser and Schoenfeld 1962; Dusart 1999; Bach and Shallit 1996).
- For every given positive  $\varepsilon$ , the interval  $(n, n + \varepsilon n)$  contains a prime, provided  $n$  is large enough.

Sometimes, for computational purposes, especially when primes in small intervals are involved, we need *effective bounds*. A classical one is Bertrand's postulate, proved in 1850 by Chebyshev:

*Chebyshev's theorem:* For every  $n > 1$  there exists a prime  $p$  such that  $n < p < 2n$ .

There are many more powerful results in the same “effective” category. We will refer to only two of them, just to give the reader a “taste”:

- For  $x \geq 25$  there is a prime  $p$  with  $x < p < 1.2x$  (Nagura 1952).
- For  $x \geq 3275$  there is a prime  $p$  with  $x < p < x \left(1 + \frac{1}{2(\ln x)^2}\right)$  (Dusart 2016).

Under a plausible but still unproven assumption (the Riemann hypothesis), the function  $\pi(x)$  allows for the sharper estimate  $\pi(x) = li(x) + O(\sqrt{x} \ln x)$  (see Davenport 2000, p. 113), where  $li(x) = \int_2^x \frac{dt}{\ln t}$  can be estimated

(see Davenport 2000, p. 54), for fixed  $q$  and  $x \rightarrow \infty$ , as  $li(x) = \frac{x}{\ln x} + \frac{1!x}{(\ln x)^2} + \dots + \frac{q!x}{(\ln x)^{q+1}} [1 + \varepsilon(x)]$ .

### 2.3.1 Harmonic Numbers Revisited

For  $n \geq 2$ , the partial sums  $H_n := \sum_{k=1}^n \frac{1}{k}$  (harmonic numbers) of the harmonic series are never integers. Having just discussed Chebyshev's theorem, it is worth noticing a nice immediate consequence. For  $n \geq 2$ , let  $p$  be the largest prime  $p \leq n$ . Then  $2p > n$  (otherwise, if  $2p \leq n$ , Chebyshev's theorem implies the existence of a prime  $q$  with  $p < q < 2p \leq n$ , contradicting the choice of  $p$ ). Thus  $\frac{1}{p}$  is the only summand in  $H_n$  that is divisible by  $p$ . That excludes the possibility of  $H_n$  being an integer, since in fraction form,  $H_n$  would have a denominator divisible by  $p$  (to exactly the first power) with a numerator not divisible by  $p$ .

Of course, the conclusion  $H_n \notin \mathbb{Z}$  for  $n \geq 2$  is traditionally reached without Chebyshev's theorem, by considering the (necessarily unique) summand  $\frac{1}{k}$  that is divisible by the highest power of 2.

But the use of Chebyshev's theorem also implies that  $H_n$  cannot be a power of order greater than 1 (square, cube, etc.) of a rational number, and also that if  $\pi(m) < \pi(n)$ , then  $H_n - H_m$  cannot be a power of order greater than 1 of a rational number.

Note that the above-mentioned refinement of Chebyshev's theorem due to Nagura (1952) can be used to prove the following proposition (Bax and Car 2004).

**Proposition 2.1** *For  $n \geq 2$ , the sum  $\sum_{k=1}^n \frac{1}{k^r}$  cannot be a perfect  $r$ -th power ( $r \geq 2$ ) of a rational number.*

*Proof* Note that Nagura's result about the existence of a prime between  $x$  and  $\frac{6}{5}x$  for  $x \geq 25$  consequently implies that for  $n \geq 11$ , there are at least two distinct primes  $p, q$  satisfying  $\frac{n}{2} < p < q < n$ . Then it is easy to see that in its reduced form,  $\sum_{k=1}^n \frac{1}{k^r}$  is necessarily of the form  $S_n := \sum_{k=1}^n \frac{1}{k^r} = \frac{A}{p^v q^q B^r}$ , with  $p, q$  not appearing in  $AB$ . If we assume  $S_n$  to be an  $r$ th power ( $r \geq 2$ ), then the exponents of all primes appearing in the reduced fraction form of  $S_n$  must be divisible by  $r$ , so that both  $p$  and  $q$  are multiples of  $r$ , a contradiction. This finishes the proof for  $n \geq 11$ . For  $n \leq 10$ , we will proceed with "brute force" computation (actually not so "brute," given the strong computational capabilities of MAPLE):

```
> for n from 2 by 1 to 10 do S(n):=sum(1/k^k,k=1..n): end do:
seq(ifactors(S(n)),n=2..10);
>
```

$$\begin{aligned} & \frac{(5)}{(2)^2}, \frac{(139)}{(2)^2(3)^3}, \frac{(8923)}{(2)^8(3)^3}, \frac{(27891287)}{(2)^8(3)^3(5)^5}, \frac{(29)(431)(60251)}{(2)^8(3)^6(5)^5}, \\ & \frac{(49739)(2153)(2617)(2213)}{(2)^8(3)^6(5)^5(7)^7}, \frac{(1173145819)(305147)(113539)}{(2)^{24}(3)^6(5)^5(7)^7}, \\ & \frac{(31)(1161244100781371)(600034207)}{(2)^{24}(3)^{18}(5)^5(7)^7}, \\ & \frac{(43)(37994429709572209)(16926029)(2441)}{(2)^{24}(3)^{18}(5)^{10}(7)^7} \end{aligned}$$

As one can easily see from the above factorizations,  $S_n$  ( $2 \leq n \leq 10$ ) is not a perfect power of a rational number of order greater than 1.

Computer algebra platforms such as MAPLE and MATLAB are widely available in colleges today. If one needs a large prime, the MAPLE function **nextprime**(**x**) will provide the smallest prime greater than  $x$  [after one has loaded MAPLE's number theory package using **with(numtheory)**]:

```
> with(numtheory):
nextprime(65145648475213541283765213825154287458542);
65145648475213541283765213825154287458651
```

Integer factorization is also straightforward in MAPLE (with **ifactors**),

```
> ifactors(87265981756178598756198561);
(3) (47) (127) (4873288756138861828123)
```

as is the determination whether a given positive integer is prime (with **isprime**),

```
> isprime(8721659812765129876519);
true
false
```

the  $n$ th prime (**ithprime**(**n**)),

```
> ithprime(100);
541
```

and the sequence of the first  $n$  primes, for example for  $n = 100$ ,

```
> seq(ithprime(k), k=1..100);
2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101,
103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193,
197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293,
307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409,
419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521,
523, 541
```

Frequently used in introductory courses in number theory or discrete mathematics are two algorithms of fundamental importance in cryptography applications:

**The extended Euclidean algorithm (EEA)** expresses the greatest common divisor  $\gcd(a, b)$  as a linear combination  $ax + by$  with  $x, y \in \mathbb{Z}$ . For example,  $\gcd(13, 11) = 1 = 13 \cdot (-5) + 11 \cdot 6$ . In MAPLE we may proceed like this (try it):

```
> igcdex(13, 11, 'x', 'y'); x; y;
```

Students may want first to get used to performing the EEA with paper and pencil, by building a table with four columns, labeled  $r$  (remainder),  $q$  (quotient),  $x$ , and  $y$ . The first two rows (besides the head) are just “data entry,” with the numbers  $a, b$  with  $a > b$  entered under the remainder column in decreasing order. Say  $a = 139, b = 49$ . The column labeled  $x$  is associated with the larger number, while the column labeled  $y$  is associated with the smaller number. At every stage, the number in the remainder column must be equal to  $a$  times the number in the  $x$  column plus  $b$  times the number in the  $y$  column. Afterward, updates will be performed using the previous two rows, as follows:

- Divide the bigger remainder by the smaller one; enter the quotient in the  $q$  column.
- Multiply the newly obtained  $q$  by the prior  $(x, y)$  vector, subtract this from the  $(x, y)$  vector obtained two steps before, and enter the result as the updated  $(x, y)$  vector.
- The numbers in the remainder column will form a decreasing sequence. Continue to update until you see zero in the remainder column. The last nonzero remainder is  $d = \gcd(a, b)$ , which will be found in addition to the desired  $x, y$  such that  $d = ax + by$ .

The EEA table below leads to  $\gcd(139, 49) = 1 = 139 \cdot 6 + 49 \cdot (-17)$ :

$r$	$q$	$x$	$y$
139	–	1	0
49	–	0	1
41	2	1	–2
8	1	–1	3
1	5	6	–17
0	...	...	...

**Fast modular exponentiation** calculates modular powers of large integers using the method of repeated squaring (symbolized by “&^”, as opposed to “^”, which would first attempt to calculate the exact integer value of the exponential (resulting in an obvious overflow error):

```
> 76456435614965&^625623456435435 mod
932485924385294387249368794238679234;
145635105502714391376863474527415053
```

Again, before we can experience the sophisticated computing power of MAPLE, it would be quite rewarding for us (both faculty and students) to have, again, a paper and pencil experience of this. A hands-on method for the fast calculation of  $a^N \bmod m$  involves a table with two columns. At the head of the table we have, in the left column,  $k$ , the “repeated squaring exponent,” which takes power of 2 values 1, 2, 4, 8, 16, . . . and continues downward for as long as  $k \leq N$ . At the head of the right column we will put  $a^k \bmod m$ . We initialize the first row with 1 ( $= k$ ) in the first column, and  $a (= a^1)$  in the second column. To update:

- In the first column, double the exponent (previous  $k$  value).
- In the second column, square the previous entry modulo  $m$ .
- Continue for as long as the item in the left column is  $\leq N$ .

Say, for example, that  $a = 23$ ,  $N = 100$ ,  $m = 541$ . The table shapes up as follows:

$k$	$a^k \bmod m$
1	23
2	529
4	144
8	178
16	306
32	43
64	226

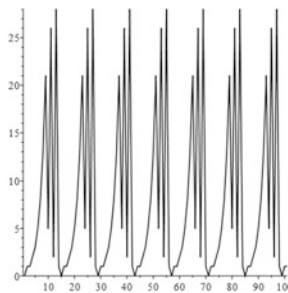
The first stage is complete. Now we have to write the exponent 100 as a sum of entries in the left column. This can be done using a base 2 representation of 100, to the effect that  $100 = 64 + 32 + 4$ . Thus  $23^{100} = 23^{64} \cdot 23^{32} \cdot 23^4 \bmod 541 = 226 \cdot 43 \cdot 144 = 366$ . Not comfortable with bases of numeration? That’s OK. We can substitute this by a “greedy” technique of writing the exponent as a sum of powers of 2, by repeatedly removing the highest power of 2 from what’s left. Say we use this for 100. The highest power of 2 not greater than 100 is 64. Put it aside and continue with the leftover, 36. The highest power of two not greater than 36 is 32. Put it aside, and we are left with a 4, which will be put aside in the next step, so we have  $100 = 64 + 32 + 4$ . It is perhaps tedious, but also rewarding and beautiful (albeit slow), but in this way we will probably appreciate even more the depth involved in such specialized computer algebra software and be able to enjoy it fully.



## 2.4 Periodic Sequences: Visualization, Periods, Preperiods, Floyd's Cycle-Finding Algorithm

The terms of the sequences investigated in this book will be defined in terms of prime numbers. When it comes to sequences, especially when one is looking for periodicity, it is useful to have a look at a plot of the sequence so that we can immediately become aware of any obvious repetitive patterns.

```
> x(0):=0; x(1):=1; for k from 2 to 100 by 1 do x(k):=(x(k-1)+x(k-2)) mod 29 end do: L:= [seq(x(k),k=0..100)]: listplot(L);
```



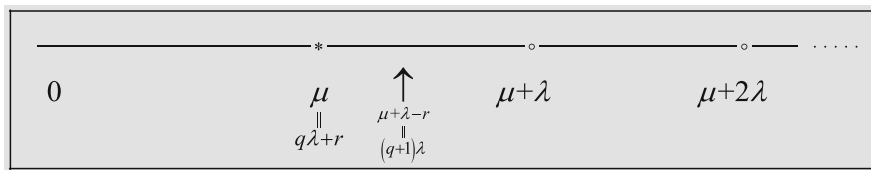
This represents a plot of the Fibonacci numbers modulo 29. The period is 14, and this can be discovered “low-tech” by gradually inspecting the picture and possibly checking neighborhoods of a few terms around subsequent repeating peaks and valleys.

This looks like an exciting adventure, but when it comes to more complex periodic sequences of integers (or integer vectors), a more formal and precise algorithm for determining the period is needed. This is Floyd’s algorithm for cycle detection, also known as *the tortoise and the hare algorithm*; cf. Knuth (1969, p. 7). This cycle-detection method is instrumental in Pollard’s rho randomized factoring algorithm (Pollard 1975).

For an ultimately periodic sequence  $x_0, x_1, x_2, x_3, x_4, \dots$  satisfying a recurrence  $x_{i+1} = f(x_i)$  with period  $\lambda$  and preperiod  $\mu$  (so that  $x_i = x_{i+\lambda}$  for all  $i \geq \mu$ , but not earlier), we proceed with two sequence calculations: a calculation at “slow” speed (also known as the *tortoise*)  $t_i = x_i, i = 0, 1, 2, \dots$ , will be synchronously produced and compared with the terms of a “fast” calculation (the *hare*)  $h_i = x_{2i}, i = 0, 1, 2, \dots$ . That is, at every step, we will compare  $x_i$  with  $x_{2i}$ . Note that the distance between the subscripts,  $i$ , that is, the distance between the hare at the front and the tortoise in the back, increases by 1 at every step.

The point of the algorithm is the following: *once both tortoise and hare enter the periodicity regime* ( $i \geq \mu$ ), due to the gradual increments, in steps of 1, of the distance between them, the distance will at some point become an integer multiple of  $\lambda$ , in which case  $x_i = t_i = h_i = x_{2i}$ . It is important to notice that the form of the

recurrence (iteration of a certain function) has as a consequence that no term of the sequence that occurs prior to the start of the cycle is equal to any one of the terms in the cycle, so that  $i \geq \mu$  implies  $x_m \neq x_n$ .



Let  $q$  and  $r$  be integers such that  $\mu = q\lambda + r$ ,  $0 \leq r < \lambda$  (keep in mind that we don't assume any a priori information about  $\lambda$  and  $\mu$ ). It is easy to see that the first index  $i$  right after  $\mu$  with the property  $x_i = x_{2i}$  is  $i = \mu + \lambda - r$  (that is,  $(q+1)\lambda$ ). At this moment when the tortoise and the hare can be construed as projecting onto the same point of the circle of circumference  $\lambda$ , we have a moment of "collision." Due to the gradual increment of the hare-tortoise distance, the next value of  $i > \mu + \lambda - r$  for which  $x_i = x_{2i}$  will occur after  $\lambda$  more steps, for  $i = \mu + 2\lambda - r = (q+2)\lambda$ , that is,  $x_{\mu+2\lambda-r} = x_{2\mu+4\lambda-2r}$ . Indeed, while the tortoise goes once around the circle, the hare does it twice, so they meet again after an increment of  $\lambda$  in the subscript: the difference between the tortoise's (that is,  $x_i$ ) subscripts  $i$  at the first two points of collision will be  $(\mu + 2\lambda - r) - (\mu + \lambda - r) = \lambda$ . Thus we obtain the period  $\lambda$ .

For the preperiod  $\mu$ , starting from the instance of the first collision ( $x_i = x_{2i}$  with  $i = \mu + \lambda - r$ ) we will go back at most  $\lambda$  times ( $\lambda$  is known at this time), more exactly  $\lambda - r$  times, and keep testing  $x_i = x_{i+q}$ ,  $x_{i-1} = x_{i-1+q}$ ,  $x_{i-2} = x_{i-2+q}, \dots$  with the last instance of continuing equality ( $x_\mu = x_{\mu+q}$ ) providing the desired preperiod  $\mu$ . Let us agree to call this particular  $i = \mu + \lambda - r$  the "tentative start" for the series of equality checks going back at most  $\lambda$  times.

Note an important byproduct of the above discussion: the "tentative start" is a multiple of the period.

Let us apply the ideas in the above discussion to concrete problems in which MAPLE will prove very useful. The first example involves a nonlinear first-order recurrent sequence in modular arithmetic (the type of sequences used in Pollard's factoring method):

*Example 2.1* Consider the sequence of integers  $\{x_n\}_{n \geq 0}$  defined as follows:

$$(*) \begin{cases} x_0 = 12345, \\ x_{n+1} = x_n^2 + 1 \pmod{4429}. \end{cases}$$

We will write down the series of MAPLE commands that lead us to the period/preperiod calculation. First of all, we will produce the "tentative start" ( $i = \mu + \lambda - r$  in the discussion above, that is, the first instance in which both tortoise and hare are in the periodicity regime, while the distance between them will be an integer multiple of the period):

```

> with(numtheory): with(ListTools): N:=12345; f:=x->x^2+3 mod N;
> x:=25: SLOW:=f(x): FAST:=f(f(x)): k:=1: while SLOW<>FAST do
SLOW:=f(SLOW): FAST:=f(f(FAST)): k:=k+1; end do:
TENTATIVE_START:=k;

```

*TENTATIVE\_START:=21*

Second, we will calculate the period by considering when tortoise and hare (now both in the cycle) meet again:

```

> x:=SLOW: SLOW:=f(x): FAST:=f(f(x)): k:=1: while SLOW<>FAST do
SLOW:=f(SLOW): FAST:=f(f(FAST)): k:=k+1; end do: PERIOD:=k;

```

*PERIOD:=7*

This means that the period is 7 and that the periodic regime starts at most 7 steps before 21.

For the verification step, the following relevant terms of the sequence will be produced:

```

> t(0):=25: for r from 1 to TENTATIVE_START+PERIOD do
t(r):=f(t(r-1)): end do:
X:=seq(t(r), r=0..TENTATIVE_START+PERIOD);

```

*X:=25, 628, 11692, 6682, 9607, 3232, 1957, 2902, 2317, 10762, 12202, 8107, 11017, 10597,  
6292, 11197, 9337, 11527, 2497, 787, 2122, 9307, 7732, 9337, 11527, 2497, 787, 2122,  
9307*

Thus the preperiod is  $\mu = 16$ , which translates into the following periodicity property of the sequence defined by  $(*) : x_{n+7} = x_n$  for every  $n \geq 16$  (and not earlier). As we shall see below, the trio “tentative start + period + preperiod” can be incorporated into a single block of MAPLE code (although for the exploration of the space of the possible periods of a certain recurrence relation, only the “period” part will be used). Before that, yet another example.

The second example will deal with second-order recurrences, more specifically the sequence of Fibonacci numbers modulo 55.

*Example 2.2* Consider the sequence of integers  $\{x_n\}_{n \geq 0}$  defined as follows:

$$(**) \quad \begin{cases} x_0 = 0, x_1 = 1, \\ x_n = x_{n-1} + x_{n-2} \pmod{55} (n \geq 2). \end{cases}$$

In order to adapt the previous calculations to second-order recurrences, we will treat a second-order recurrence as a first-order recurrence, albeit with 2D vectors:

$$(x_n, x_{n+1}) \rightarrow (x_{n+1}, x_n + x_{n+1} \pmod{55})$$

```

> with(numtheory): with(ListTools): f:=(x,y)->(y,x+y mod 55);
      f:=(x,y)->(y,(x+y) mod 55)
> X:=(0,1): SLOW:=f(X): FAST:=f(f(X)): k:=1: while SLOW<>FAST do
SLOW:=f(SLOW): FAST:=f(f(FAST)): k:=k+1: end do:
TENTATIVE_START:=k;
      TENTATIVE_START:=20
> X:=SLOW: SLOW:=f(X): FAST:=f(f(X)): k:=1: while SLOW<>FAST do
SLOW:=f(SLOW): FAST:=f(f(FAST)): k:=k+1: end do: PERIOD:=k;
      PERIOD:=20
> t(0):=0: t(1):=1: for r from 2 to TENTATIVE_START+PERIOD do
t(r):=t(r-1)+t(r-2) mod 55 end do:
X:=[seq(t(r),r=0..TENTATIVE_START+PERIOD)];
      X:=[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 0, 34, 34, 13, 47, 5, 52, 2, 54, 1, 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 0,
      34, 34, 13, 47, 5, 52, 2, 54, 1, 0]

```

In this case, the conclusion is that the Fibonacci sequence modulo 55 has period 20 with a preperiod of 0.

Faced with the case of a particular recurrent sequence  $x_n = f(x_{n-1}, \dots, x_{n-d})$  of a certain order  $d$ , with the initial conditions constituting the “seed” vector  $SEED := (x_0, x_1, \dots, x_{d-1})$  already in place, and keeping in mind that the “tentative start” is an integer multiple of the period, as observed previously, we shall use the following compact block of MAPLE code to derive all three, in that order:

- Tentative start
- Period
- Preperiod

We have seen above how to produce the first two. The idea for the preperiod is to start with the vectors ( $d$ -tuples), namely the initial  $d$ -tuple  $U := "SEED"$  and  $V$  (the  $d$ -tuple found after a shift of “tentative start” ahead), setting a variable  $m$  with an initial value of zero, proceeding with the shifting of the above-mentioned  $d$ -tuples  $U$  and  $V$ , one step at a time, via the process

$$(x_k, x_{k+1}, \dots, x_{k+d-1}) \mapsto (x_{k+1}, \dots, x_{k+d-1}, f(x_{k+d-1}, x_{k+d-2}, \dots, x_{k+1}, x_k)),$$

while increasing  $m$  by 1 for each shift, *for as long as  $U$  and  $V$  remain distinct*. This will continue until we get to the situation  $U = V$  (which must happen, and when it happens for the first time,  $m = PERIOD$ ). Let’s take, for example, the third-order recurrence

$$\begin{cases} x_n = P(x_{n-1} + x_{n-2} + x_{n-3}), \\ x_0 = 11, x_1 = 3, x_2 = 37. \end{cases}$$

where  $P(x)$  is the greatest prime factor of  $x$  (an important function for our purposes, which will become one of our heroes a little bit later; the essential thing needed here is that there are reasons to believe that a whole class of recurrence relations similar to this one produce ultimately periodic sequences).

```

> f:=(x1,x2,x3)->(x2,x3,P(x3+x2+x1)): SEED:=(11,3,37): X:=SEED:
SLOW:=f(X): FAST:=f(f(X)): k:=1: while SLOW<>FAST do
SLOW:=f(SLOW): FAST:=f(f(FAST)): k:=k+1: end do:
TENTATIVE_START:=k: Y:=SLOW: X:=SLOW: SLOW:=f(X): FAST:=f(f(X)):
k:=1: while SLOW<>FAST do SLOW:=f(SLOW): FAST:=f(f(FAST)):
k:=k+1: end do: PERIOD:=k: m:=0: U:=SEED: V:=Y: while U<>V do
m:=m+1: U:=f(U): V:=f(V) end do: PREPERIOD:=m;

```

We get, fairly quickly,

```

TENTATIVE_START := 400
PERIOD := 100
PREPERIOD := 346

```

Again, let's say that we don't believe it until we see it (which is natural for amateur programmers like the author). Let's get enough terms:

```

> x(0):=11: x(1):=3: x(2):=37: for r from 3 to 500 do
x(r):=P(x(r-1)+x(r-2)+x(r-3)): end do: L:=seq(x(r),r=0..500):

```

Then by directly selecting suitable blocks of three consecutive terms, we indeed obtain the confirmation of the Floyd algorithm's above "triple output":

$$(x_{346}, x_{347}, x_{348}) = (5, 13, 7) = (x_{346+100}, x_{347+100}, x_{348+100}),$$

$$(x_{345}, x_{346}, x_{347}) = (17, 5, 13) \neq (31, 5, 13) = (x_{345+100}, x_{346+100}, x_{347+100}).$$

In many cases, especially when we will investigate (through a sort of "Monte Carlo–Floyd" hybrid) the distribution of the periods with initial conditions randomly varied, we are interested in the period only. This means that after specifying the recurrence form and the seed, we will use only the first part of the code that leads to the period:

```

> X:=SEED: SLOW:=f(X): FAST:=f(f(X)): k:=1: while SLOW<>FAST
do SLOW:=f(SLOW): FAST:=f(f(FAST)): k:=k+1: end do:
TENTATIVE_START:=k: Y:=SLOW: X:=SLOW: SLOW:=f(X): FAST:=f(f(X)):
k:=1: while SLOW<>FAST do SLOW:=f(SLOW): FAST:=f(f(FAST)):
k:=k+1: end do: PERIOD:=k;

```

**Exercise:** assume that after the "tentative start" and the "period" are identified, somebody proposes the following MAPLE procedure (based rather on "backtracking"):

```

> PP:=proc(X::list, TS::integer, P::integer);
> k:=TS;
> while X[k]-X[k+P]=0 do
> k:=k-1;
> end do;
> PP:=k;
> end proc;

```

Assume that somebody computes and stores the list

```

> X:=[seq(x(r), r=0..TENTATIVE_START+PERIOD)]:

```

Does the MAPLE instruction “PP(X, TENTATIVE\_START, PERIOD)” provide the preperiod? When does it fail?

## 2.5 Mathematical Beauty at the Addition/Multiplication Interface

The sequence of natural numbers  $1, 2, 3, 4, 5, 6, \dots$  ( $a_n = n$ , the OEIS sequence A000027) is archetypal. Ideas involving the natural numbers evolved and became increasingly sophisticated over millennia, beginning with ancient tally marks and proceeding through the emergence of the sexagesimal numeration system in Mesopotamia.

With addition, the set of natural numbers  $\mathbb{N} = \{1, 2, 3, 4, 5, 6, \dots\}$  is a semi-group  $(\mathbb{N}, +)$  generated by a single element, indeed  $\mathbb{N} = \langle 1 \rangle$ . Here is a method to analyze the “speed” of additive generation that will be applied later in the context of other special binary operations. Starting from  $A_0 = \{1\}$ , we define an ascending sequence of sets  $\{A_n\}_{n \geq 0}$  through the recurrence  $A_{n+1} = A_n \cup \{x+y \mid x, y \in A_n\}$ .

That is, at every step, the current set  $A_n$  is enlarged with all the values of the binary operation with arguments in  $A_n$ . It is not difficult to see that at each step, the set size doubles, i.e.,  $A_n = \{1, 2, 3, \dots, 2^n\}$ . We clearly have  $A_0 \subset A_1 \subset A_2 \subset \dots$  and  $\langle 1 \rangle = \bigcup_{n=0}^{\infty} A_n = \mathbb{N}$ .

As we shall see, if we replace addition with another well-chosen binary operation “ $\circ$ ” on  $\mathbb{N}$  and repeat the above procedure for the structure  $(\mathbb{N}, \circ)$ , with sets  $\{A_n\}_{n \geq 0}$  defined accordingly, the study of the speed of generation (specifically the limit  $\lim(|A_{n+1}|/|A_n|)$ ) will lead us to an amazing conjecture that will provide a description of the golden section  $(1 + \sqrt{5})/2$  in purely arithmetic terms.

In themselves, addition and multiplication on  $\mathbb{N}$  are “easy.” The prime factorization structures of two natural numbers  $a, b$ , if known (granted, that’s a big “if,” since factoring integers is a computationally hard problem), make the prime factorization of their product  $ab$  immediately available. Indeed, the prime factorization theorem reduces the multiplication  $ab$  to additions of the corresponding exponents of the various prime powers appearing in  $a, b$ : if  $a = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k}$  and  $b = p_1^{\beta_1} p_2^{\beta_2} \dots p_k^{\beta_k}$  with  $\alpha_i, \beta_i \geq 0$ , then  $ab = p_1^{\alpha_1 + \beta_1} p_2^{\alpha_2 + \beta_2} \dots p_k^{\alpha_k + \beta_k}$ . In “fancier” terms,

that can be alternatively formulated by saying that due to the prime factorization theorem, the multiplicative semigroup of positive integers is isomorphic to the direct sum of a countable family of identical additive monoids of the form  $(\mathbb{N}_0, +)$ , where  $\mathbb{N}_0 = \{0, 1, 2, 3, 4, 5, \dots\}$ .

The really interesting problems appear, though, at the interface between additive and multiplicative aspects, a zone that can be rightfully considered the *Mariana Trench* of number theory. Through the process of integer factorization itself, an interesting connection was revealed (Billingsley 1973) with Brownian motion.

We can start by thinking that there is no immediate formula for the prime factorization of  $a + b$  in terms of those of  $a$  and  $b$ . Going only a little deeper within that interface will quickly lead us to difficult problems that are still open. If we look at the prime numbers as the pinnacles of the *multiplicative* universe, what happens when we start to *add* primes:  $2 + 2 = 4$ ,  $3 + 3 = 6$ ,  $3 + 5 = 8$ ,  $3 + 7 = 10$ ,  $5 + 7 = 12$ , and so on? It doesn't take long to notice the strong likelihood that every even number that is greater than 2 is a sum of two primes. This is the celebrated Goldbach conjecture, still unproven since it was stated on June 7, 1742, when Christian Goldbach formulated the conjecture in a letter to Leonhard Euler (Goldbach 2016).

Note that although it is important to be aware of various kinds of asymptotic behavior involving “primes at the additive–multiplicative interface,” the proofs for most of them involve advanced analytic number theory and are beyond the scope of this book, which is limited to elementary approaches. That said, among many interesting problems emerging at this interface, we will mention the following.

The Sophie Germain primes, that is, primes  $p$  such that  $2p + 1$  is also a prime (OEIS sequence A005384, 2, 3, 5, 11, 23, 29, 41, 53, 83, 89, 113, 131, 173...). A heuristic result (Shoup 2009, p. 123) estimates the number of Sophie Germain primes less than or equal to  $x$  to be a constant multiple of  $x/(\ln x)^2$ . A similar estimate exists for the number of primes  $p$  less than or equal to  $x$  such that  $p + 2$  is also prime, i.e.,  $(p, p + 2)$  form a twin prime pair. It has been proved (Brun's theorem, 1919) that the sum of the reciprocals of the primes in the twin pairs is convergent. The conjectural infinitude of both Sophie Germain primes and twin prime pairs is still unproven.

Primes in arithmetic progressions constitute another amazing phenomenon emerging at the additive–multiplicative interface. One of the most important number-theoretic results proved recently is the celebrated Green–Tao theorem (Green and Tao 2008), to the effect that there are arbitrarily long arithmetic progressions consisting of primes. The longest known such progression (discovered in 2010), due to Benoît Perichon, consists of 26 primes (see OEIS sequence A204189). As an example on the asymptotics side, in a seminal 2010 paper (Green and Tao 2010), Green and Tao estimated the number of four-term arithmetic prime progressions  $1 < p_1 < p_2 < p_3 < p_4 \leq N$  as  $(1 + o(1)) \cdot \frac{3}{4} \prod_{p \geq 5} \left(1 - \frac{3p-1}{(p-1)^3}\right) \frac{N^2}{\log^4 N}$ , or approximately  $\frac{0.4764 \cdot N^2}{\log^4 N} (1 + o(1))$ .

On a related note, it is important to notice here the classical result (Dirichlet's theorem) on the existence of infinitely many primes in an arithmetic progression  $\{qn + a | n = 0, 1, 2, \dots\}$ , where  $\gcd(a, q) = 1$ ; see Apostol (1976, p. 148), Hardy and Wright (1979, p. 13). The asymptotic estimate for the number  $\pi(x; q, a)$  of primes in the above progression that are less than or equal to  $x$  is, as one would intuitively expect,  $\pi(x; q, a) \approx \frac{1}{\phi(a)} \frac{x}{\ln x}$  (Davenport 2000, p. 121). The sharper estimate  $\pi(x; q, a) \approx \frac{li(x)}{\phi(q)} + O(x^{1/2+\varepsilon})$  holds under a stronger assumption (the generalized Riemann hypothesis).

We can generalize Sophie Germain pairs to Cunningham chains (of the first kind), which are sequences of primes of the form  $p, 2p+1, 4p+3, 8p+7, \dots, 2^{k-1}p + 2^{k-1} - 1$ . For example, 2, 5, 11, 23, 47 is such a Cunningham chain of length 5. A similar heuristic argument applies to estimating the number of primes  $p$  less than or equal to  $x$  such that  $p$  is the first term in a first-order Cunningham chain of length  $n$  as a certain constant (expressed as an infinite product over the primes) multiple of  $x/(\ln x)^n$ .

This can be further extended by Dickson's conjecture (Dickson 1904), which claims that given a finite set of linear forms  $L_i(x) = a_i x + b_i$ ,  $1 \leq i \leq n$ , there are infinitely many integer values of  $k$  such that  $L_i(k)$  is prime for all  $i$  with  $1 \leq i \leq n$  (unless obvious existing divisibility relations prevent that). The distribution of such  $k$  is assumed to obey the same estimate of the form of a constant multiple of  $x/(\ln x)^k$ . More generally, "Schinzel's hypothesis H" (Schinzel and Sierpinski 1958) asserts a similar result for polynomials of arbitrary degree, that is, if  $f_i(x) \in \mathbb{Z}[x]$ ,  $1 \leq i \leq n$ , are polynomials with positive leading coefficients, then there are infinitely many integer values of  $k$  such that  $f_i(k)$  is prime for  $1 \leq i \leq n$ , unless an obvious condition prevents that.

The sequences (of integers, sets, or arrays) that will be discussed in this work will be positioned at the same conjecture-rich interface.

## 2.6 Some Classical Recurrent Sequences. Ducci Games

For a recurrent sequence of order  $k$ , the first  $n$  terms (making up "the seed")  $x_0, x_1, \dots, x_{k-1}$  are given, together with a recurrence relation of the form  $x_n = f(x_{n-1}, x_{n-2}, \dots, x_{n-k})$ , which holds for  $n \geq k$ . In MAPLE we can quickly list a desired number of terms in a recurrent sequence. For example, the first 20 terms in the "tribonacci" recurrent sequence  $\{t_n\}_{n \geq 0}$  defined as  $t_0 = 0, t_1 = 1, t_2 = 1, t_n = t_{n-1} + t_{n-2} + t_{n-3}$  for  $n \geq 3$  can be obtained as follows:



```
> t(0):=0: t(1):=1: t(2):=1: for r from 3 to 19 do
> t(r):= t(r-1)+t(r-2)+t(r-3) end do:
> seq(t(r),r=0..19);
```

0, 0, 1, 1, 2, 4, 7, 13, 24, 44, 81, 149, 274, 504, 927, 1705, 3136, 5768, 10609, 19513, 35890

But of course, the Fibonacci sequence  $(F_n)_{n \geq 0}$  (OEIS A000045) is arguably the most celebrated among the classical sequences:

$$\begin{cases} F_0 = 0, F_1 = 1, \\ F_n = F_{n-1} + F_{n-2} \text{ for } n \geq 2. \end{cases}$$

The history of the Fibonacci numbers can be traced back to ancient Sanskrit musical theory (Singh 1985). In the European space, they first appear in Fibonacci's 1202 book *Liber Abaci* [see the translation by Sigler (2002)].

MAPLE has a specific package with a built-in function for the Fibonacci numbers:

```
> with(combinat, fibonacci): seq(fibonacci(k),k=0..30);
```

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, 514229, 832040

In MATLAB we can use the Fibonacci recurrence form to produce a Fibonacci function:

```
function fibo = fibo(n)
f(1)=1;
f(2)=1;
for I=3:n;
    f(I)=f(I-1)+f(I-2);
end;
fibo=f(n);
```

The remarkable fact that the ratios of consecutive Fibonacci numbers converge to the golden section,

$$\lim_{n \rightarrow \infty} \frac{F_{n+1}}{F_n} = \frac{1 + \sqrt{5}}{2}$$

was first noticed by Johannes Kepler in his 1611 “Essay on the Six-Cornered Snowflake” (see Colin Hardie’s translation (Hardie 1966).

An explicit formula for the Fibonacci numbers in terms of the golden section and its conjugate was known to De Moivre and derived by Binet (see Livio 2002, p. 108):

$$F_n = \frac{1}{\sqrt{5}} \left[ \left( \frac{1 + \sqrt{5}}{2} \right)^n - \left( \frac{1 - \sqrt{5}}{2} \right)^n \right].$$

In connection to modular arithmetic, an interesting sequence consists of the Pisano periods or the periods of the Fibonacci numbers modulo  $n$ . See Mark Renault's thesis (Renault 1996) and OEIS A001175 (<http://oeis.org/A001175>), where the following nice MAPLE program for the sequence of Pisano periods, due to Alois P. Heinz, can be found:

```
> a:= proc(n) local f, k, l; l:= ifactors(n)[2];
>     if nops(l)<>1 then ilcm(seq(a(i[1]^i[2]), i=1))
>     else f:= [0, 1];
>         for k do f:=[f[2], f[1]+f[2] mod n];
>             if f=[0, 1] then break fi
>         od; k
>     fi
> end:
> seq(a(n), n=1..100);
```

1, 3, 8, 6, 20, 24, 16, 12, 24, 60, 10, 24, 28, 48, 40, 24, 36, 24, 18, 60, 16, 30, 48, 24, 100, 84, 72,  
48, 14, 120, 30, 48, 40, 36, 80, 24, 76, 18, 56, 60, 40, 48, 88, 30, 120, 48, 32, 24, 112, 300,  
72, 84, 108, 72, 20, 48, 72, 42, 58, 120, 60, 30, 48, 96, 140, 120, 136, 36, 48, 240, 70, 24,  
148, 228, 200, 18, 80, 168, 78, 120, 216, 120, 168, 48, 180, 264, 56, 60, 44, 120, 112, 48,  
120, 96, 180, 48, 196, 336, 120, 300

Numerous Fibonacci identities can found online at Ron Knott's multimedia website on Fibonacci numbers (Knott 2016). Among them is the Cassini identity  $F_{n-1}F_{n+1} - F_n^2 = (-1)^n$ . Also, every issue in the journal *Fibonacci Quarterly* has a set of interesting problems in both elementary and advanced categories. It is well known that Fibonacci numbers can be expressed as diagonal sums in Pascal's triangle [for this and other patterns in Pascal's triangle see Bogomolny (2016)].

If we change the initial conditions from 0, 1 (as is the case for the Fibonacci numbers) to 2, 1, we get another interesting integer sequence: the Lucas numbers  $(L_n)_{n \geq 0}$  (OEIS A000032):

$$\begin{cases} L_0 = 2, L_1 = 1, \\ L_n = L_{n-1} + L_{n-2} \text{ for } n \geq 2. \end{cases}$$

The analogue of Binet's formula is

$$L_n = \left( \frac{1 + \sqrt{5}}{2} \right)^n + \left( \frac{1 - \sqrt{5}}{2} \right)^n.$$

Like the Fibonacci numbers, the Lucas numbers have interesting combinatorial connections. It is fairly well known that if we assume periodic boundary conditions, the number of bit strings of length  $n$  without consecutive 1's is  $L_n$ . Using the transfer matrix method (Gessel and Stanley 1995, Section 7), the number of such arrangements can be represented as the trace of the  $n$ th power of the transfer matrix

$$T = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \text{ with eigenvalues } \lambda_{1,2} = (1 \pm \sqrt{5})/2, \text{ so that } L_n = \text{tr}(T^n) = \lambda_1^n + \lambda_2^n.$$

Similarly, one can see that if we don't assume periodic boundary conditions, the number of bit strings of length  $N$  without consecutive 1's is the Fibonacci number  $F_{n+2}$ .

Interesting probabilistic analogues of Fibonacci numbers have been studied (Heyde 1980; Viswanath 1999; Emb and Tref 1987).

In other types of recurrences, a vector with numerical components is expressed in terms of the previous vector  $X_{k+1} = f(X_k)$  for  $k \geq 0$ . An interesting such recurrence is the "Ducci game," or the " $N$ -number game" (Chamb and Thomas 2004), apparently originating in the late 1800s and named after Professor E. Ducci (Honsberger 1970; Furno 1981). In this kind of vector recurrence, if  $X_k = (x_1, x_2, \dots, x_{N-1}, x_N) \in \mathbb{Z}^N$ , then the vector following  $X_k$  will be

$$X_{k+1} = (|x_1 - x_2|, |x_2 - x_3|, \dots, |x_{N-1} - x_N|, |x_N - x_1|).$$

It is well known that if  $N$  is a power of 2, then the above iteration eventually leads to the null  $N$ -tuple  $(0, 0, \dots, 0, 0)$ . Otherwise, the Ducci iteration may eventually enter a limit cycle of length greater than 1 with the interesting feature that for a fixed integer  $C > 0$ , every vector in the limit cycle has all components in the set  $\{0, C\}$  (which makes the Ducci map essentially a binary iteration). Detailed data on cycle lengths of  $N$ -number Ducci games for  $N$  up to 40 are provided in Calkins et al. (2005). The 37-number Ducci game is listed with two possible cycle lengths: 1 and 3,233,097 (the largest cycle size for the Ducci games considered in the study). To see an example in which the number of components in a Ducci vector is not a power of two, let's use MAPLE to play a "5-number game" starting from  $(7, 5, 6, 7, 1)$ . Note that after 10 iterations, we eventually enter a cycle of length 15. The arrays below display the preperiod and period (the 10th vector and the 25th vector coincide), with the first column playing the role of a counter for the iterative step in the Ducci game.

```

> a(1,1):=0: a(1,2):=7: a(1,3):=5: a(1,4):=6: a(1,5):=7:
a(1,6):=1: N:=26: for k from 2 by 1 to N do a(k,1):=k-1:
a(k,2):=abs(a(k-1,2)-a(k-1,3)): a(k,3):=abs(a(k-1,3)-a(k-1,4)):
a(k,4):=abs(a(k-1,4)-a(k-1,5)): a(k,5):=abs(a(k-1,5)-a(k-1,6)):
a(k,6):=abs(a(k-1,6)-a(k-1,2)): end do:
interface(rtablesize=infinity): g:=(i,j)->a(i,j): Matrix(N,6,g);

```

0	7	5	6	7	1	10	1	0	0	0	1	18	1	1	1	0	1
1	2	1	1	6	6	11	1	0	0	1	0	19	0	0	1	1	0
2	1	0	5	0	4	12	1	0	1	1	1	20	0	1	0	1	0
3	1	5	5	4	3	13	1	1	0	0	0	21	1	1	1	1	0
4	4	0	1	1	2	14	0	1	0	0	1	22	0	0	0	1	1
5	4	1	0	1	2	15	1	1	0	1	1	23	0	0	1	0	1
6	3	1	1	1	2	16	0	1	1	0	0	24	0	1	1	1	1
7	2	0	0	1	1	17	1	0	1	0	0	25	1	0	0	0	1
8	2	0	1	0	1												
9	2	1	1	1	1												

For the 4-number game, W.A. Webb has proved (Webb 1982) that the games in which it takes the longest time to get to the null vector are (essentially) those starting from an initial four-tuple consisting of consecutive tribonacci numbers  $(t_n, t_{n-1}, t_{n-2}, t_{n-3})$ , in which case the Ducci iteration takes  $3\lfloor \frac{n}{2} \rfloor$  steps until it reaches the null cycle. For example, let's use MAPLE to see what happens if we start from  $(t_{13}, t_{12}, t_{11}, t_{10}) = (927, 504, 274, 149)$ .

0	927	504	274	149	10	48	24	160	88
1	423	230	125	778	11	24	136	72	40
2	193	105	653	355	12	112	64	32	16
3	88	548	298	162	13	48	32	16	96
4	460	250	136	74	14	16	16	80	48
5	210	114	62	386	15	0	64	32	32
6	96	52	324	176	16	64	32	0	32
7	44	272	148	80	17	32	32	32	32
8	228	124	68	36	18	0	0	0	0
9	104	56	32	192					

For completeness let us briefly sketch here a “common-sense” proof of the fact that every four-number Ducci game eventually reaches the null state  $(0, 0, 0, 0)$ . Three main elementary ideas are to be considered, with the third being the “punch line” consequence of the first two.

First, the entries in the 4D vectors appearing during the Ducci iteration are bounded. If one such vector is  $(a, b, c, d)$  with  $0 \leq a, b, c, d \leq M$ , then the entries occurring in the subsequent vector  $(|a - b|, |b - c|, |c - d|, |d - a|)$  are still in the interval  $[0, M]$ .

Second, it is easy to see that if we reduce the Ducci iteration  $(a, b, c, d) \mapsto (|a - b|, |b - c|, |c - d|, |d - a|)$  modulo 2, it becomes particularly simple. The “mod 2 Ducci” iteration is  $(\alpha, \beta, \gamma, \delta) \mapsto (\alpha + \beta, \beta + \gamma, \gamma + \delta, \delta + \alpha)$ . At this point, a direct verification shows that after four such modular iterations, the mod 2 Ducci iteration arrives at the null modular vector (check this!). Going back to integers, this means that an additional factor of two emerges for the entries in a vector after every block of four integer Ducci iterations. Consequently, the vectors appearing in the Ducci game are divisible by higher and higher powers of 2.

Thirdly, it is easy to see that putting together the two facts about the integer entries that “they are bounded” and “they become divisible by increasing powers of 2” leads to the conclusion that the null integer vector is eventually reached (indeed, if an integer in the interval  $[0, M]$  becomes divisible by a power of 2 with  $2^k > M$ , it must necessarily equal 0).

The limiting binary behavior of the  $N$ -number Ducci iteration makes it equivalent, if we focus on the cycle only, to multiplication by  $1 + x$  in the ring  $F_2[x]/(x^N - 1)$ . This leads to a comfortable placement of the Ducci problem in the area of cyclotomy (Calkins et al. 2005; Breuer et al. 2007). An interesting connection with Artin’s conjecture is established (Breuer et al. 2007), to the effect that if  $p$  is prime and 2 is a primitive root modulo  $p$ , then the  $p$ -number Ducci games have only one possible period other than 1.

Numerous other variations and generalizations of the Ducci sequences theme have been considered: analogues in higher dimensions (Breuer 2007), Ducci sequences over the reals and their asymptotic behavior (Brockman and Zerr 2007; Brown and Merzel 2003), Ducci sequences with algebraic numbers (Caragiu et al. 2011), the dynamics of Ducci sequences defined in terms of various weightings (Chamberland 2003),  $p$ -adic Ducci games (Car and Bax 2007), etc. A very interesting Ducci-type special iteration is introduced in Cobeli and Zaharescu (2014), where the authors play with the “atomic transformation function”  $Z(a, b) = \frac{ab}{(\gcd(a, b))^2}$  (as a consequence, the authors state and prove a variation of Gilbreath’s conjecture).

The Ducci game modulo 2 constitute a special case of a one-dimensional *cellular automaton* (Berto and Tagliabue 2012) encoded as “rule 102” in Wolfram’s classification (Weisstein). The evolution of cellular automata reveals interesting patterns of emerging complexity and self-organization, which makes them potentially useful in the study of complex physical systems and statistical mechanics (Wolfram 1983). In the same conceptual family lies the celebrated “Game of Life” due to John Conway (Gardner 1970). Later on, we will consider a series of variations on the cellular automaton concept, with the help of special number-theoretic functions.

An infinite-dimensional Ducci game analogue, lying at the above-mentioned additive–multiplicative interface, is the classical Gilbreath conjecture in number theory (Caldwell). Invented in 1958 by napkin-doodling computer expert and

amateur magician Norman O. Gilbreath (*Genii Magazine*), this is about successively generating a string of infinite sequences using a Ducci-type iteration starting with the sequence of primes  $p_1, p_2, p_3, p_4, \dots, p_k, p_{k+1}, p_{k+2}, \dots$ . Every sequence  $x_1, x_2, x_3, x_4, \dots, x_k, x_{k+1}, x_{k+2}, \dots$  that occurs in the process is updated as

$$|x_1 - x_2|, |x_2 - x_3|, |x_3 - x_4|, |x_4 - x_5|, \dots, |x_k - x_{k+1}|, |x_{k+1} - x_{k+2}|, |x_{k+2} - x_{k+3}|, \dots$$

Gilbreath's conjecture asserts that the first entry of each such sequence, with the exception of the first one, equals 1.

2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59	61	67	71...
1	2	2	4	2	4	2	4	6	2	6	4	2	4	6	6	2	6	4	2...
1	0	2	2	2	2	2	2	4	4	2	2	2	2	0	4	4	2	2...	
1	2	0	0	0	0	0	2	0	2	0	0	0	2	4	0	2	0...		
1	2	0	0	0	0	2	2	2	2	0	0	2	2	4	2	2...			
1	2	0	0	0	2	0	0	0	2	0	2	0	2	2	0...				
1	2	0	0	2	2	0	0	2	2	2	2	2	0	2...					
1	2	0	2	0	2	0	2	0	0	0	0	2	2...						
1	2	2	2	2	2	2	2	0	0	0	2	0...							
1	0	0	0	0	0	0	2	0	0	2	2...								
1	0	0	0	0	0	2	2	0	0	0...									
1	0	0	0	0	2	0	2	0	0...										

.....

The second line in Gilbreath's table is the sequence of prime gaps (OEIS sequence of differences between consecutive primes, A001223). This brings us to the following elementary classical situation.

Once of the nicest elementary results about primes, to the effect that there are arbitrarily large gaps between consecutive primes, uses the beautiful idea of the compositeness of the terms of the arithmetic progression

$$n! + 2, n! + 3, \dots, n! + n.$$

Indeed, these are  $n - 1$  integers divisible by  $2, 3, \dots, n$ , respectively.

As for computational evidence, in 1993 Andrew Odlyzko verified the Gilbreath conjecture up to a row rank of  $3.4 \cdot 10^{11}$  (Odlyzko 1993).

## 2.7 Deeper into the Randomness

Everything we care about lies somewhere in the middle, where pattern and randomness interlace.—James Gleick, *The Information: A History, a Theory, a Flood*

Generating mathematical randomness is a paradox and an art. Random number generation is important for computer security, while random structures and related randomized algorithms are crucial elements in the contemporary computing environment. A great way to introduce students to higher mathematics is to let them study the  $\pm 1$  sequences of Legendre symbols  $\left(\frac{1}{p}\right), \left(\frac{2}{p}\right), \dots, \left(\frac{p-1}{p}\right)$  describing the distribution of quadratic residues and nonresidues modulo a prime  $p$ . The study of the patterns of quadratic residues and nonresidues is an extremely interesting problem that attracted considerable attention in the period ranging from the end of the nineteenth century into the first half of the twentieth century (Aladov 1896; Davenport 1931; Davenport 1933; Peralta 1992). Ultimately, this distribution problem turns out to be intimately connected to one of the “big problems” in mathematics, the Riemann hypothesis for curves over finite fields, settled in 1949 by Weil (1949), generalizing a “genus 1” result obtained by Helmut Hasse that estimates the number of points  $N$  of an elliptic curve  $y^2 = x^3 + ax + b$  over the finite field with  $p$  elements as  $|N - (p + 1)| \leq 2\sqrt{p}$ .

Students can quickly get hands-on computational experience using MAPLE. For example, if  $p = 19$ , the  $\pm 1$  sequence of Legendre symbols can be quickly obtained as follows:

```
> with(numtheory):
> p:=19: L:= [seq(legendre(k,p), k=1..p-1)];
      L := [1, -1, -1, 1, 1, 1, 1, -1, 1, -1, 1, -1, -1, -1, 1, 1, -1]
```

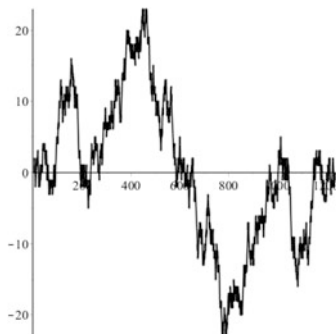
It is not hard (for both faculty and students) to become fascinated by this mix of global order and local randomness. Even the “smallest” problems that arise in the process are likely to have deep connections with analytic number theory. For example, just reflecting on the rank of the appearance of the first  $-1$  in the  $\pm 1$  sequence generated, as above, for arbitrary primes  $p$  leads to the deep problem of the “least quadratic nonresidue” (Burgess 1957). The OEIS sequence listed as A053760 presents the smallest quadratic nonresidue modulo the  $n$ th prime for  $n = 1, 2, 3, \dots$ . The first “unusually high” smallest quadratic nonresidue appears to be the one modulo  $p = 311$ . Indeed, using MAPLE as above, it will be easy to verify that in the sequence  $\left(\frac{1}{311}\right), \left(\frac{2}{311}\right), \dots, \left(\frac{310}{311}\right)$ , the first  $-1$  appears only at the eleventh position.

By incorporating MAPLE packages (in addition to **numtheory**, we will be needing **plots** and **ListTools**), it is always illuminating to visualize the sequence of

partial sums  $\left(\frac{1}{p}\right) + \left(\frac{2}{p}\right) + \cdots + \left(\frac{k}{p}\right)$  with  $1 \leq k \leq p-1$ . For example, let us use  $p = 1237$ , a prime congruent to 1 modulo 4. This will lead us to the following globally symmetric, locally random display for the sequence of partial sums.

```
> with(numtheory): with(plots): with(ListTools):
> p:=1237; L:= [seq(legendre(k,p),k=1..p-1)]: S:=PartialSums(L):
listplot(S);
```

$p := 1237$



The globally symmetric shape follows from the multiplicative property of the Legendre symbol modulo the particular prime  $p = 1237$ , to the effect that  $\left(\frac{-x}{1237}\right) = \left(\frac{-1}{1237}\right)\left(\frac{x}{1237}\right) = \left(\frac{x}{1237}\right)$ . The locally random character follows, for example, from the Pólya–Vinogradov inequality (Davenport 2000), which, adapted to the case of the Legendre symbol, consists of the partial character sum estimate

$$\left| \sum_{k=N+1}^{N+H} \left(\frac{k}{p}\right) \right| \leq \sqrt{p} \ln p.$$

In order to get rid of global symmetry and get plots more and more resembling one-dimensional random walks, one has the option of using Legendre symbols with a polynomial argument. For example, the use of a separable cubic polynomial  $f(x) = x^3 + ax + b$  over the finite prime field  $F_p$  leads us to the sets

$$X(a, b, p) = \{x \in F_p \mid x^3 + ax + b \text{ is a square}\} \subseteq F_p.$$

The subsets  $X(a, b, p) \subset F_p$  behave like “random” subsets with a density of (roughly) 0.5. We can define related “elliptic walks” (see Caragiu et al. 2006), where the agreement was to replace the possible zero values of the quadratic characters with 1’s with the following specifications:



$$W_{a,b,p}(k) = \begin{cases} 1, & \text{if } f(k) \text{ is a square in } F_p, \\ -1, & \text{if } f(k) \text{ is a nonsquare in } F_p, \end{cases}$$

$$B_{a,b,p}(k) = \sum_{j=1}^k W_{a,b,p}(j), \quad k = 1, 2, \dots, p.$$

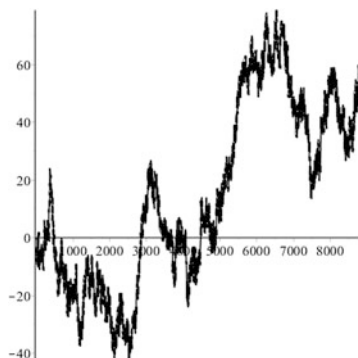
It was proved in Caragiui et al. (2006) through chi-square statistical testing that the above-mentioned “elliptic walks” cannot be distinguished from genuine random walks if we use criteria such as the number of returns to the origin (*the expected number of returns* to the origin for a random  $\pm 1$  walk of length  $2n$  returning to the origin exactly  $r$  times is  $\frac{1}{2^{2n-r}} \binom{2n-r}{r}$  according to (Feller 1968, p. 96), the grouping test, or the runs test (Knuth 1969, p. 74, Ex. 14).

However, if other criteria are used, such as the total displacement that for “elliptic walks” is confined (Caragiui et al. 2006) to the interval  $[-2\sqrt{p}, 2\sqrt{p}]$  according to the Hasse–Weil estimate, then elliptic walks are distinguishable from genuine random walks. Indeed, for  $\lambda < \mu$ , the probability that the endpoint of a truly symmetric  $\pm 1$  random walk of length  $p$  is in the interval  $[\lambda\sqrt{p}, \mu\sqrt{p}]$  equals (Feller 1968, p. 76), as a consequence of the central limit theorem,  $\frac{1}{\sqrt{2\pi}} \int_{\lambda}^{\mu} e^{-\frac{x^2}{2}} dx$ , so that as a consequence, a fraction of  $\frac{1}{\sqrt{2\pi}} \int_3^4 e^{-\frac{x^2}{2}} dx = 0.00131822\dots$  symmetric  $\pm 1$  random walks of any length  $p$  have their endpoints in  $[3\sqrt{p}, 4\sqrt{p}]$ , a condition that is not satisfied by the elliptic walks defined as above.

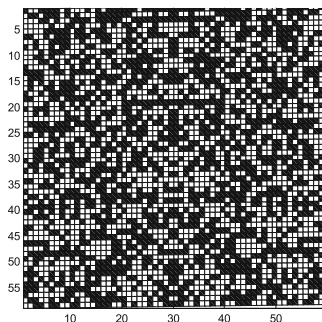
In any case, a useful class exploration of (pseudo)random structures will lead students to use MAPLE to visualize the incomplete character sums (which coincide with “elliptic walks” with the few possible exceptions where  $j^3 + aj + b = 0$ ):

$$\sum_{j=1}^k \left( \frac{j^3 + aj + b}{p} \right), \quad k = 1, 2, \dots, p-1.$$

```
> p:=8837; L:= [seq(legendre(k^3+3*k+1,p),k=1..p-1)]:
S:=PartialSums(L): listplot(S);
```



We can easily use MAPLE to visualize two-dimensional random structures. One possibility is to construct a two-dimensional analogue of elliptic walks by considering cubic polynomials over a finite field of the form  $F_{p^2}$  with the prime  $p$  being subject to  $p \equiv 3 \pmod{4}$ , a constraint used for convenience due to the relation  $F_{p^2} = F_p[i] = \{x + iy | x, y \in F_p\}$ , which helps in visualizing  $F_{p^2}$  as a  $p \times p$  rectangular grid. The image below represents a projection (obtained by MATLAB) of the elliptic curve  $y^2 = x^3 + x + 3$  over the base field  $F_{59^2}$ , that is, a pictorial description of all perfect squares in the field  $F_{p^2}$ .



NOTE: A more general procedure is to use a two-variable polynomial  $f(x, y) \in F_p[x, y]$  with coefficients in an arbitrary finite prime field  $F_p$  and display in a similar manner the points  $(x, y) \in F_p \times F_p$  such that  $f(x, y)$  is a perfect square in  $F_p$ . Using various averaging/smoothing procedures (such as replacing the value at each node of the grid with the average over a certain neighborhood), we can get similar displays in color.

Later on, we will introduce new such “models of randomness” using the greatest prime factor function and Conway’s subprime function. But first, we will briefly provide an overview of functions that will play an important role in our future constructions.

## 2.8 The Greatest Prime Factor Function

For a positive integer  $x$ , the greatest prime factor function  $P(x)$  is the largest prime factor of  $x$ , with the proviso that  $P(1) := 1$ . This will be the main function that we will be using in our sequential experiments. The OEIS entry on the greatest prime factor (A006530) was always particularly inspiring in our work on greatest prime factor sequences. It features a concise MAPLE function dealing with the greatest prime factor function due to Peter Luschny, where **factorset** (producing the set of all prime factors) is followed by *op*, which presents this set as a numerical list for which we are able to calculate the **max**:

```
> with(numtheory):
> P:=n->max(1,max(op(factorset(n))));
```

In MATLAB we can use the direct procedure

```
function gpf=gpf(n)
gpf=max(factor(n));
end
```

Over the years, the greatest prime factor function has received substantial attention from the standpoint of advanced analytic number theory, and many outstanding results and estimates have been obtained. Finding asymptotic estimates for the greatest prime factor function is an enterprise that goes a long way back. In 1930, K. Dickman investigated the probability  $P(x, n)$  that a randomly selected integer  $k \in \{1, 2, \dots, n\}$  satisfies  $p := P(k) < n^x$ , showing that the expected value of  $x$  such that the random variable  $p$  is of the form  $p = n^x$  is  $0.62432999\dots$ , which came to be known as the *Golomb–Dickman constant* (Dickman 1930; Weisstein; Knuth and Pardo 1976).

In Kemeny (1993), it was proved that the average value of  $\frac{P(n)}{n}$  is, asymptotically,  $\frac{\zeta(2)}{\ln n}$ , while the probability that  $P(n) > \sqrt{n}$  equals  $\ln 2 = 0.69314718\dots$  in the asymptotic limit.

To list a few other examples from the fertile and engaging work on the greatest prime factor function, Erdős showed (1952) that if  $f(x) \in \mathbb{Z}[x]$  is a polynomial that does not split as a product of linear polynomials in  $\mathbb{Z}[x]$ , and if  $P_x := P\left(\prod_{k=1}^x f(k)\right)$ , then for some  $c > 0$ , the estimate  $P_x > x(\ln x)^{c \ln \ln x}$  holds. In fact, there are numerous very interesting results on the greatest prime factor of the product of consecutive integers. In Laishram and Shorey (2005) an elementary proof is provided to the effect that  $P(n(n+1)\dots(n+k-1))$  is greater than  $2k$  if  $n > \max(k+13, (279/262)k)$ , and it is greater than  $1.97k$  if  $n > k+13$ .

Hooley investigated the greatest prime factor of a quadratic polynomial (Hooley 1967). Grytczuk et al. showed (2001) that the Fermat numbers  $F_m = 2^{2^m} + 1$  satisfy  $P(F_m) \geq 2^{m+2}(4m+9) + 1$  for  $m \geq 4$ . For different types of numbers, one of the

results proved by Erdős and Shorey in (1976) implies that  $P(2^p - 1) \gg p \ln p$  for all primes  $p$ . Luca and Najman (2011) provided the solution of the inequality  $P(x^2 - 1) < 100$  in natural numbers.

In the previous classical intermezzo, we showed how factorials are used to build sequences of consecutive composite integers. With the greatest prime factor concept introduced, the following proposition of a similar flavor can be proved. These kinds of results at the borderline between “easy” and “wow” make for excellent tools and supplements for the undergraduate number theory classroom:

**Proposition 2.2** *Let  $M > 0$  be fixed. Then for every positive integer  $k$  one can find a sequence of consecutive integers  $n + 1, n + 2, \dots, n + k$  such that  $P(n + r) > M$  for  $r = 1, 2, \dots, k$ .*

*Proof* Let  $p_1, p_2, \dots, p_k$  be distinct prime numbers greater than  $M$ . With the Chinese remainder theorem (Leveque 1977, p. 60), one can infer the existence of a positive integer  $n$  such that the following congruences are satisfied:

$$\begin{aligned} n &\equiv -1 \pmod{p_1} \\ n &\equiv -2 \pmod{p_2} \\ &\dots\dots\dots \\ n &\equiv -k \pmod{p_k} \end{aligned}$$

Then for  $1 \leq r \leq k$ , we have  $P(n + r) \geq p_r > M$ , which concludes the proof.

## 2.9 Overview of Some Other Number-Theoretic Functions and Sequences

In what follows, we will briefly review some other important functions, besides the greatest prime factor, that will be used in other experiments to be conducted in our “sequential laboratory.”

### (A) The least prime factor function

The least prime factor function  $p(n)$  is indexed by OEIS as A020639:

1, 2, 3, 2, 5, 2, 7, 2, 3, 2, 11, 2, 13, 2, 3, 2, 17, 2, 19, 2, 3, 2, 23, 2, 5, 2, 3, 2, 29, 2, 31, 2, 3.

In relation to the greatest prime factor  $P(n)$ , Erdős and van Lint (1982) proved the following asymptotic series result:  $\sum_{n \leq x} \frac{p(n)}{P(n)} = \frac{x}{\ln x} + \frac{3x}{\ln^2 x} (1 + o(1))$  as  $x \rightarrow \infty$ .

On a truly elementary note, the next intermezzo is recognized as a constant inhabitant of virtually all undergraduate classes that involve discussions on factoring, e.g., discrete mathematics, abstract algebra, introductory number theory, fundamental mathematics for teachers.

If  $n > 1$  is a composite integer, then the least prime factor of  $n$  is less than or equal to  $\sqrt{n}$ , with equality if  $n$  is the square of a prime.

The least prime factor of an integer  $n$  is recorded by OEIS as A020639, where the following MAPLE routine (attributed to R.J. Mathar) is recorded:

```
> LPF:= proc(n) if n = 1 then 1; else
min(op(numtheory[factorset](n))) ; end if; end proc;
```

In MATLAB we can use the following:

```
function lf=lf(n)
lf=min(factor(n));
end
```

### (B) Conway's subprime function

Related to the least prime factor function, Conway's "subprime function" is defined as

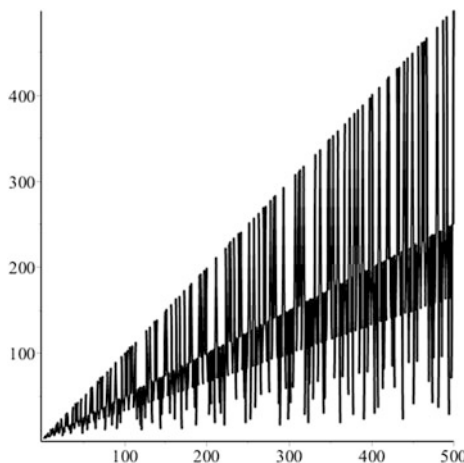
$$C(x) = \begin{cases} x, & \text{if } x \text{ is prime or } x = 1, \\ x/LPF(x), & \text{otherwise.} \end{cases}$$

Alternatively,  $C(1) := 1$ , and otherwise,  $C(n)$  is the largest proper divisor of  $n$ . The function  $C(x)$  gained prominence in relation to a class of interesting analogues of the Fibonacci recurrence, namely Conway's subprime Fibonacci sequences; see the 2012 Tatiana Khovanova blog entry (Khovanova 2012) and the 2014 *Mathematics Magazine* paper (Guy et al. 2014). More about Conway's subprime function, with related sequences and nonassociative structures, can be found in Chapter 4.

The MAPLE code used for the Conway function together with a plot of the list of the values  $C(n)$ ,  $1 \leq n \leq 500$  is shown below.

```
> C:=proc(n::integer) local u: if isprime(n)='true' or n=1 then
n else u:=factorset(n): n/min(seq(u[j], j=1..nops(u))) end if
end proc;
> listplot([seq(C(n), n=1..500)]);
```

Note that the first three clearly identifiable slopes in the list plot above correspond to primes, even numbers, and odd multiples of 3.



### (C) Euler's Phi Function

We have to mention the ubiquitous Euler's totient function (see Chapter 6 in Andrews 1994), or Chapter 5 in Hardy and Wright (1979), which can be expressed in terms of the prime factors  $p$  of  $n$  as follows:

$$\varphi(n) = n \prod_{p|n} \left(1 - \frac{1}{p}\right).$$

This function (which can also be seen as the number of invertible elements in  $\mathbb{Z}/n\mathbb{Z}$  or the number of generators of the additive group of integers modulo  $n$ ) is one of the jewels of number theory, which has found amazing application in high-end modern cryptosystems such as RSA. A consequence of enormous importance of the totient function being the cardinality of the group of invertible elements is Euler's theorem.

**Euler's Theorem** *For every  $a$  with  $\gcd(a, n) = 1$  we have  $a^{\varphi(n)} \equiv 1 \pmod{n}$ .*

We will be using this theorem to build and investigate a mysterious analogue of the Fibonacci sequence.

If  $p$  is a prime, then the number of primitive roots modulo  $p$  is  $\varphi(p-1)$ . A nice related result is the following.

**Proposition 3** *Let  $\varepsilon > 0$ . Then there are infinitely many primes  $p$  such that the probability that a randomly chosen element in  $F_p^*$  is a primitive root modulo  $p$  is less than  $\varepsilon$ .*

*Proof* From Mertens's third theorem, we have the asymptotic relation  $\prod_{p \leq x} \left(1 - \frac{1}{p}\right) \sim \frac{e^{-\gamma}}{\ln x}$  (see Section 22.8 in Hardy and Wright 1979), where  $\gamma = 0.577215664\dots$  is the Euler–Mascheroni constant. Consequently, we can find a finite set of primes  $q_1, q_2, \dots, q_k$  such that  $\prod_{i=1}^k \left(1 - \frac{1}{q_i}\right) < \varepsilon$ . Using Dirichlet's theorem, we can see that there are infinitely many primes  $p$  such that  $p \equiv 1 \pmod{p_1 p_2 \dots p_k}$ . For every such prime, we have  $\varphi(p-1) = (p-1) \prod_{q|p-1} \left(1 - \frac{1}{q}\right) \leq (p-1) \prod_{1 \leq i \leq k} \left(1 - \frac{1}{q_i}\right) < \varepsilon(p-1)$ . So the probability that a randomly chosen element in  $F_p^*$  is a primitive root modulo  $p$  is  $\frac{\varphi(p-1)}{p-1} < \varepsilon$ .

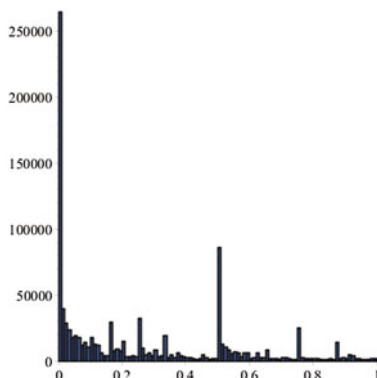
And now an interesting byproduct of this. In conjunction with a strong model-theoretic result (Chatzidakis et al. 1992), the above elementary proposition plus the fact that  $\lim_{p \rightarrow \infty} \varphi(p-1) = \infty$  (in general,  $\varphi(n) \geq \sqrt{\frac{n}{2}}$ , Delanoy) has as a consequence that there is no formula  $\Phi(x)$  in the first-order language of rings such that for every prime  $p$ , the set of primitive roots modulo  $p$  coincides with the set of elements in the finite field  $F_p$  validating  $\Phi$ , that is, the primitive roots are not first-order definable (Caragiu 2000).

#### (D) Lehmer's totient problem: a statistical experiment

We have seen that for a prime  $p$  we have  $\varphi(p) = p-1$ , but whether there are composite numbers  $n$  with  $\varphi(n)$  a divisor of  $n-1$  is still an open question, known as “Lehmer's totient problem.” Numerous computational investigations have been made; for example, every potential solution  $n$  must be greater than  $10^{20}$  and have at least 14 prime divisors (Cohen and Hagis 1980), and for every potential solution  $n$  that is divisible by 3, it must be the case that  $n$  is greater than  $10^{360,000,000}$  and has at least 40 million prime divisors (Burcsi et al. 2011).

A simple exploratory MAPLE-based exercise around the aforementioned Lehmer's problem may consider visualizing the statistical distribution of the fractional parts of the numbers  $\frac{n-1}{\varphi(n)}$  in order to witness a possible unevenness that would tilt the distribution toward zero. To this end, we will visualize the histogram of the fractional parts  $\left\{\frac{n-1}{\varphi(n)}\right\}_{n=1}^{1,000,000}$ . Here is what we get:

```
> with(numtheory): with(Statistics):
> L:= [seq( evalf( frac( (n-1)/phi(n) ) ), n=1..1000000) ]: Histogram(L,
frequency scale=absolute, bincount=100);
```



These computer algebra visuals have the potential of becoming pivotal educational moments that will inspire undergraduates to become involved with further research in number theory: in their journey, this could be the first step. Notice a major “accumulation point” toward zero and a minor accumulation point toward the middle, 0.5.

#### (E) The $3x + 1$ (Collatz) problem

Erdős offered \$500 for the solution to this problem, while offering the sobering assessment to the effect that “*Mathematics may not be ready for such problems*” (Lagarias 1985). It states that for every seed  $x_0$ , the sequence defined by the recurrence

$$x_{n+1} = \begin{cases} 3x_n + 1, & \text{if } x_n \text{ is odd,} \\ \frac{x_n}{2}, & \text{if } x_n \text{ is even.} \end{cases}$$

eventually reaches 1 (and hence enters the cycle 1, 4, 2). Serious computing power has been invested in the problem, with many articles published featuring generalizations of Collatz’s problem. When it comes to formal results, Krasikov and Lagarias (2003) proved that for all sufficiently large values of  $x$ , at least  $x^{0.84}$  integers  $S$  from 1 to  $x$  have the property that the Collatz iteration with seed  $x_0 = s$  eventually reaches 1. Collatz’s problem has many connections. Particularly interesting are the those (Kontorovich and Miller 2005; Lagarias and Soundararajan 2006) with Benford’s law (Hill 1996) for the distribution of the leading digit in numerous real-life data sets.

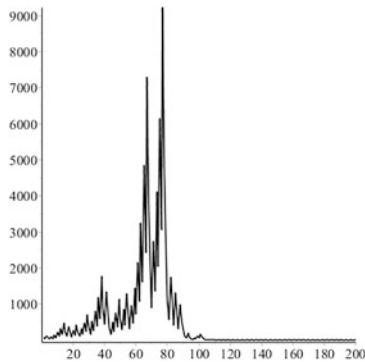
An unusually long preperiod in Collatz’s iteration occurs for  $x_0 = 27$  (see OEIS’s A008884,  $3x + 1$  sequence starting at 27). The following MAPLE set of instructions (Monagan) can be used to obtain a view on the process, a bumpy ride toward the limit cycle.



```

> with(plots):
> x[0]:= 27;
> for i from 0 to 200 do:
> if x[i] mod 2=0 then x[i+1] := x[i]/2:
> else x[i+1] := 3*x[i]+1 fi;
> od;
> L:= [seq(x[i], i=1..200)]: listplot(L);

```



In MATLAB, this can be achieved by

```

function collatz = collatz(a,n)
x(1)=a;
for I=2:n;
    if mod(x(I-1),2)==0;
        x(I)=x(I-1)/2;
    else
        x(I)=3*x(I-1)+1;
    end
end
collatz=plot(x);

```

The On-Line Encyclopedia of Integer Sequences has a wealth of entries (714 as of November 2016) in reference to Collatz's problem, with several involving prime numbers, e.g., A078350 (the number of primes in the Collatz iteration starting with  $n$ ), A070975 (the number of steps required to get to 1 if one starts from the  $n$ th prime), etc.

## 2.10 MATLAB Too!

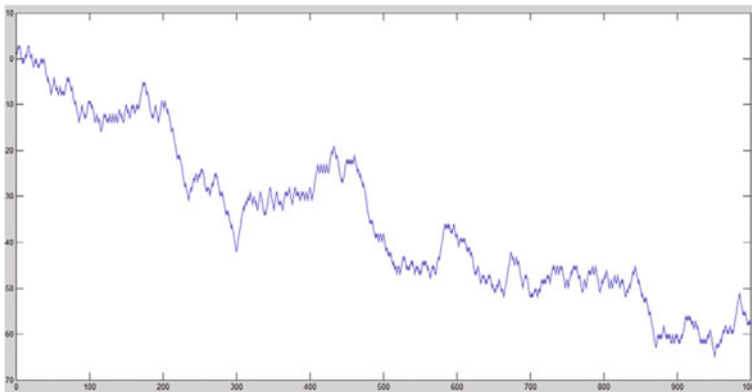
We have been speaking mostly about MAPLE, but MATLAB is equally useful: we love them both! However, the author and his students at the time found custom-made functions somehow easier to write in MATLAB (although probably if we had known more about MAPLE, that opinion would have changed). We would like to share a few of these functions that were either used or attempted in senior capstone projects.

### • The Blum–Blum–Shub random number generator

This is a random number generator (see Blum et al. 1986) of importance to cryptography, which uses a modulus  $M = pq$  with  $p, q$  large primes. It calculates terms  $x_i$  using the recurrence  $x_{n+1} = x_n^2 \pmod{N}$  from an initial “seed”  $x_0 = a$ , and at every step takes  $x_i \bmod 2 \in \{0, 1\}$  as a term in the generated pseudorandom bit string. For the purpose of improved visualization, we will instead work with the associated  $\pm 1$  strings (just use the function  $x \mapsto 2x - 1$  to the effect  $0 \mapsto -1, 1 \mapsto 1$ ) and plot their cumulative sums.

```
function bbs = bbs(p,q,n,a)
m=p*q;
x(1)=a;
z(1)=2*mod(a,2)-1;
for I=2:n;
    x(I)=mod(x(I-1)^2,m);
    z(I)=2*mod(x(I),2)-1;
end
y=cumsum(z);
bbs=plot(y)
```

For example, the choice  $p = 983, q = 967$  with  $a(=x_0) = 103$  and  $N = 1000$  steps will produce the following image for the associated  $\pm 1$  walk:



### ➤ Functions related to the Euclidean algorithm

The following MATLAB function calculates the number of divisions in the Euclidean algorithm (also see OEIS entry A051010)

```

function euclid = euclid(a,b)
h=max(a,b);
l=min(a,b);
I=0;
while l ~= 0;
    I=I+1;
    r=mod(h,l);
    h=l;
    l=r;
end;
euclid=I;

```

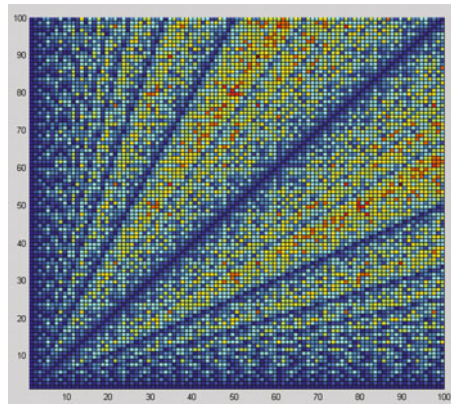
Lamé's theorem asserts that the number of divisions in the Euclidean algorithm is at most five times the number of digits of the smaller number, with the maximum attained for pairs of consecutive Fibonacci numbers. If you want to get a nice color plot displaying these numbers over an  $n \times 1$  grid, use the following function.

```

function euclidmat = euclidmat(n)
for I=1:n;
    for J=1:n;
        x(I,J)=euclid(I,J);
    end;
end;
euclidmat=pcolor(x)

```

For example, **euclidmat(1000)** will produce the following display:



As for the obviously visible edges in the image, the “valleys” (minima) corresponds to the pairs of numbers that are multiples of each other, while the “ridges” (maxima) are related (as their slopes in the image) to the quotients of consecutive Fibonacci numbers.

The next function produces a plot displaying the evolution (exponential decay) of the sequence of successive remainders in a particular Euclidean algorithm.

```

function euclidgraph = euclidgraph(a,b)
h=max(a,b);
l=min(a,b);
I=0;
while l ~= 0;
    I=I+1;
    x(I)=1;
    r=mod(h,l);
    h=l;
    l=r;
end;
euclidgraph=plot(x);

```

### ➤ Pollard's rho

Randomized factoring using the polynomial  $x^2 + a$ ; note that students are encouraged to experiment with their “favorite” functions)

```

function rhotest = rhotest(n,a)
x=2;
y=2;
d=1;
J=0;
while d==1;
    x=rem(x^2+a,n);
    y=rem((y^2+a)^2+a,n);
    d=gcd(abs(x-y),n);
    J=J+1;
end
if d==n;
    rhotest=0
else
    rhotest=d;
    J
end
end

```

### ➤ Prime walks

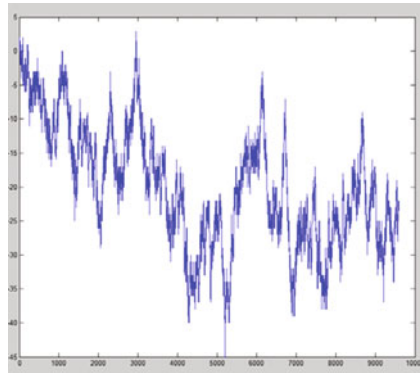
Choose a sequence of odd primes 3, 5, 7, 11, 13, 17, .... To each term we associate  $+1$  or  $-1$  depending on whether the prime is congruent to 1 or 3 modulo 4. The sequence of cumulative sums will give us a view of the interwoven distribution of primes in the two congruence classes modulo 4.

```

function [primewalk]=primewalk(m)
x=primes(m);
k=size(x,2);
for I=2:k;
    x(I)=2-mod(x(I),4);
end;
y=cumsum(x);
primewalk=plot(y);

```

For example, **primewalk(100000)** will produce the following:



### ➤ A computational exercise with a “Collatz-type” generation

The following is a problem in the category of “startups,” that is, new problems that are interesting enough that we may conjecture that they will attract some attention in the future. That said, let us consider the following function inspired by Collatz’s recurrence:

$$Coll(x, y) = \begin{cases} (x+y)/2 & \text{if } x \equiv y \pmod{2}, \\ 3(x+y) + 1, & \text{otherwise.} \end{cases}$$

It may be seen as a binary operation on the set of natural numbers. This MATLAB exercise will engage students in the exploration of the structure generated by an initial “seed set”  $A$  by recursively applying the Collatz operation starting from  $X_0 = A$ . We are thus looking at the set recurrence

$$X_{n+1} = X_n \cup \{Coll(x, y) | x, y \in X_n\}.$$

The MATLAB function for *Coll* is

```
function coll = coll(x, y)
if mod(x+y, 2) == 0;
    coll = (x+y) / 2;
else
    coll = 3*x + 3*y + 1;
end
end
```

The set generation will be performed by the following function:

```

function [groupoidcollatz] =
groupoidcollatz(x)
N=size(x,2);
K=N;
y=x;
for I=1:N;
    for J=1:N;
        a=coll(I,J);
        K=K+1;
        y(K)=a;
    end
end
groupoidcollatz=sort(distinct(y));
end

```

So let's begin! Note that if we start with  $A = \{1\}$ , then since  $\text{Coll}(1, 1) = 1$ , all the subsequent iterates will equal  $\{1\}$ . Let us work then with the seed  $X_0 = A = \{2\}$ . In this case, the first iteration leads to  $X_1 = \{2, 7\}$ :

```

>> groupoidcollatz(2)

ans =

    2    7

```

The second iteration leads to  $X_2 = \{2, 7, 17, 29\}$ , the third iteration leads to  $X_3 = \{2, 7, 17, 19, 23, 29, 31, 37, 59, 61, 79, 89, 137, 167\}$ , and the fourth iteration gives an 85-element set:

$X_4 = \{2, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 107, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 199, 211, 227, 241, 251, 257, 269, 277, 293, 307, 313, 331, 337, 347, 353, 359, 367, 373, 383, 409, 419, 421, 439, 443, 449, 467, 479, 487, 499, 503, 509, 569, 577, 593, 599, 617, 641, 647, 659, 719, 727, 761, 857, 1049\}$ .

This will be rapidly followed by higher-order generating sets with cardinalities increasing very rapidly:

$$|X_5| = 616, |X_6| = 3875, |X_7| = 24,061, \dots$$

Clearly, the set of natural numbers cannot be covered in this way, since, for example, the element 1 cannot be written as  $\text{Coll}(x, y)$  with  $x, y \in \mathbb{N}$  appearing at a previous level unless  $x = y = 1$ .

A myriad of questions opens for exploration at this point, for the excitement of both students and faculty participating in the project. What happens, then, if we change the seed set to  $A = \{1, 2\}$ ? To  $A = \{1, 2, 4\}$ ? Can we find instances with a reasonable description of  $\bigcup_{n=0}^{\infty} X_n$ ? Which primes will eventually fall into one of the  $X_n$ ? The sky is the limit!

## 2.11 An Experiment with Pairs of Primitive Roots Modulo Primes

In classical number theory, a great deal of attention has been paid to certain special pairs of primitive roots modulo a prime number, for which either

- the sum is 1; we shall call these “Category 1” pairs  $(\alpha, \beta)$  of primitive roots modulo  $p$  such that  $\alpha + \beta = 1$ , or
- the difference is 1; we shall call these “Category 2” pairs  $(\alpha, \alpha + 1)$  of primitive roots.

We can use MAPLE to quickly find the least primitive root modulo a prime  $p$ , traditionally denoted by  $g_p$ . The sequence of least primitive roots modulo the  $n$ th prime is listed in OEIS as A001918. MAPLE can quickly produce an initial segment:

```
> gp:=n->pprimroot(ithprime(n));
> seq(gp(n),n=1..100);
1, 2, 2, 3, 2, 2, 3, 2, 5, 2, 3, 2, 6, 3, 5, 2, 2, 2, 7, 5, 3, 2, 3, 5, 2, 5, 2, 6, 3, 3, 2, 3, 2, 6, 5, 2, 5, 2, 2,
2, 19, 5, 2, 3, 2, 3, 2, 6, 3, 7, 7, 6, 3, 5, 2, 6, 5, 3, 3, 2, 5, 17, 10, 2, 3, 10, 2, 2, 3, 7, 6, 2, 2, 5, 2,
5, 3, 21, 2, 2, 7, 5, 15, 2, 3, 13, 2, 3, 2, 13, 3, 2, 7, 5, 2, 3, 2, 2
```

Grosswald provided an estimate for the size of the least primitive root, proving that  $g_p < p^{0.499}$  for all large enough  $p$  (Grosswald 1981). As an important existence result, it has been proved (Moreno and Sotero 1990) that if  $q > 2$ , then the finite field  $F_q$  contains two (not necessarily distinct) primitive elements  $\alpha, \beta$  such that  $\alpha + \beta = 1$ .

Note that there is a direct connection between the Category 1 pairs of primitive roots and the so-called “Costas arrays” (the “Golomb construction”) with applications to radar and sonar systems (Costas 1984; Cohen-Mullen 1991).

Regarding Category 2 pairs of primitive roots, a conditional result, due to Vegh, shows that if  $p$  is a prime greater than 3 such that  $\varphi(p)/(p-1) > 1/3$ , then there exists at least one Category 2 pair of primitive roots modulo  $p$  (Vegh 1968). Later, S.D. Cohen showed (Cohen 1985) unconditionally that every finite field  $F_q$  with  $q > 3$ ,  $q \not\equiv 7 \pmod{12}$ ,  $q \not\equiv 1 \pmod{60}$  contains a pair of consecutive primitive roots.

It is easy to see that there are Category 2 pairs modulo 2, 3, and 7. If  $\mathbf{C}$  is the set of prime powers  $q$  such that there exists a pair of primitive roots of the form  $\alpha, \alpha + 1$  in the finite field  $F_q$ , the following is conjectured by S.D. Cohen in Cohen (1985):

**Conjecture** (Cohen 1985) *The set  $\mathbf{C}$  contains all prime powers except 2, 3, and 7.*

We will now experiment a little with “small” special pairs of primitive roots. First of all, let us agree on the following definition of “smallness” for such pairs: for Category 1 pairs,  $\alpha, \beta$  (primitive roots with sum 1, where  $\alpha \leq \beta$ ), we will define

“small” to signify that  $\alpha < \sqrt{p}$ . That is, they are pairs of the form  $k, p+1-k$  with  $k < \sqrt{p}$ . For Category 2 pairs  $\alpha, \alpha+1$  (primitive roots with difference 1), we will again define “small” to signify that  $\alpha < \sqrt{p}$ .

A conjecture regarding these “small pairs” was formulated together with my student J.C. Schroeder in the context of his senior research (Schroeder 2013), done in 2013:

**(CSPPR) Conjecture on Small Special Pairs of Primitive Roots** For all large enough primes  $p$ , one can find small special pairs of primitive roots of Category 1 (with sum 1) as well as small special pairs of primitive roots of Category 2 (with difference 1).

Finding computational support for the CSPPR conjecture was exciting, especially the work on a “lab manual” showing the data collection performed during that period.

A preliminary analysis involves the first 16 primes. In these cases, the existence of such small pairs generally does not hold, with the exception of the prime 29, which allows for both the Category 1 pair (3, 27) and the Category 2 pair (2, 3) Table 2.1.

As we continue, we notice that gradually more and more cases appear in which both Category 1 and Category 2 special pairs of primitive roots do exist.

**Table 2.1** Very small primes

Prime $p$	Category 1 (sum 1) Small (least $< \sqrt{p}$ )	Category 2 (difference 1) Small (least $< \sqrt{p}$ )
2	no	no
3	no	no
5	no	(2, 3)
7	no	no
11	no	no
13	no	no
17	no	no
19	no	(2, 3)
23	no	no
29	(3, 27)	(2, 3)
31	no	no
37	no	no
41	no	no
43	no	no
47	(5, 43)	no
53	(3, 51)	(2, 3)



Prime $p$	Category 1	Category 2
59	(6, 54)	<i>no</i>
61	(7, 55)	(6, 7)
67	(7, 61)	<i>no</i>
71	(7, 65)	<i>no</i>
73	<i>no</i>	<i>no</i>
79	(3, 77)	(6, 7)
83	(5, 79)	(5, 6)
89	(7, 83)	(6, 7)
97	<i>no</i>	<i>no</i>
101	(3, 99)	(2, 3)
103	(5, 99)	(5, 6)
107	(5, 103)	(5, 6)
109	<i>no</i>	(10, 11)
113	(6, 108)	(5, 6)
127	<i>no</i>	(6, 7)
131	(6, 126)	<i>no</i>
137	(6, 132)	(5, 6)
139	<i>no</i>	(2, 3)
149	(3, 147)	(2, 3)
151	(6, 146)	(6, 7)
157	(6, 152)	(5, 6)
163	(11, 153)	(11, 12)
167	(5, 163)	<i>no</i>
173	(3, 171)	(2, 3)
179	(6, 174)	(6, 7)
181	<i>no</i>	<i>no</i>
191	<i>no</i>	<i>no</i>
193	<i>no</i>	<i>no</i>
197	(3, 195)	(2, 3)
199	(3, 197)	<i>no</i>
211	(7, 205)	(2, 3)
223	(10, 214)	(5, 6)
227	(5, 223)	(13, 14)
229	(7, 223)	(6, 7)
233	(6, 228)	(5, 6)

The trend towards fewer instances of “no” continues; here are the data for the primes from 1967 to 1999, the last double “no” appearing for the prime 1873:

Prime $p$	Category 1	Category 2
1867	(12, 1856)	(12, 13)
1871	(14, 1858)	(41, 42)
1873	<b>no</b>	<b>no</b>
1877	(12, 1866)	(11, 12)
1879	(19, 1861)	(11, 12)
1889	(7, 1883)	(6, 7)
1901	(3, 1899)	(2, 3)
1907	(5, 1903)	(5, 6)
1913	(6, 1908)	(5, 6)
1931	(8, 1924)	(13, 14)
1933	(15, 1919)	(14, 15)
1949	(3, 1947)	(2, 3)
1951	(3, 1949)	(41, 42)
1973	(3, 1971)	(2, 3)
1979	(6, 1974)	(17, 18)
1987	(5, 1983)	(2, 3)
1993	(21, 1973)	(20, 21)
1997	(3, 1995)	(2, 3)
1999	(6, 1994)	(29, 30)

We notice progressively a larger density of primes with small special pairs of primitive roots of both categories. Indeed, every prime starting from the 2362nd to the 10,000th displays both Category 1 and Category 2 small pairs of primitive roots. To see this, the approach was to introduce, for every prime  $p$ , the functions

$$\min\{2p - 2 - \text{ord}(p - k + 1) - \text{ord}(k)\}_{k=2.. \lfloor \sqrt{p} \rfloor}$$

and

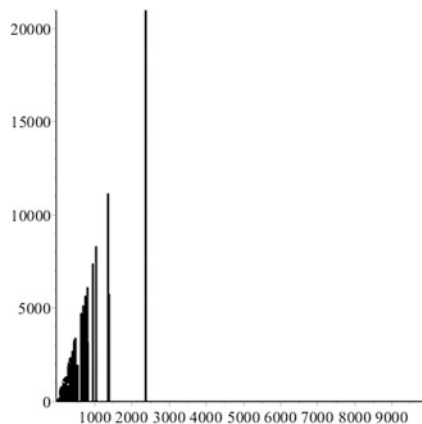
$$\min\{2p - 2 - \text{ord}(k) - \text{ord}(k + 1)\}_{k=2.. \lfloor \sqrt{p} \rfloor}.$$

Note that these are both zero (or equivalently, their sum is zero) precisely if  $p$  has a small Category 1 pair of primitive roots and also a small Category 2 pair of primitive roots. Consequently, the MAPLE function

```
> S:=r->min(seq(2*ithprime(r)-2-order(ithprime(r)-
k+1,ithprime(r))-
order(k,ithprime(r)),k=2..floor(sqrt(ithprime(r)))))+min(seq(2*
ithprime(r)-2-order(k,ithprime(r))-
order(k+1,ithprime(r)),k=1..floor(sqrt(ithprime(r)))));
```

is zero at  $r$  precisely when the  $r$ th prime allows for both types of small pairs of primitive roots. A plot of the list of the values  $S(r)$  will reveal the “zero regions” corresponding to the primes with both types of small pairs of primitive roots, thus allowing us to lean toward the plausibility of the CSPPR conjecture.

```
> L:=seq(S(r),r=5..10000): listplot(L);
```



It is conceivable that the CSPPR conjecture follows in even stronger forms from far-reaching hypotheses like the generalized Riemann hypothesis, which would imply a “power-of-the-logarithm”  $O(\ln^C p)$  upper bound for the least primitive root modulo  $p$ .

## 2.12 Traffic Flow and Quadratic Residues

Traffic flow theory began to emerge as a sophisticated area of application of mathematics in the second half of the twentieth century (Mannering and Kilareski 1990; May 1990). It is a huge area, but we will be concerned with only a particularly simple cellular automaton (CA) (Weisstein) model for one-dimensional traffic flow, the CA “rule 184” or “traffic rule” (Rosenblueth and Gershenson 2011; Wentian 1987). This CA model was the topic of a senior research course in 2010 at Ohio Northern University (Brace 2010). Here we plan to use it to visualize what happens if perfect squares modulo a prime  $p$  are assimilated to “vehicles” riding on an imaginary “highway.”

Under rule 184, a CA iteration consists in making any cell hosting a 1 advance one step to the right, provided the cell sitting in front of it hosts a 0. If we have only one cell hosting a 1 (“the vehicle”) and otherwise only zeros, then the 1 will continue moving at a constant speed to the right. In general, we have more vehicles

on the highway. Say we have 10 cells (with periodic boundary conditions and that begin moving rightward) from the following initial state:

1	1	0	1	0	0	1	0	1	0
---	---	---	---	---	---	---	---	---	---

Then the next couple of states will be as follows:

1	0	1	0	1	0	0	1	0	1
0	1	0	1	0	1	0	0	1	1
1	0	1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0	1	0

We see that this leads to a cycle of length 2. In general, though, the traffic patterns are very complex and nonlinear.

As a nice elementary exercise illustrating a connection with Fibonacci and Lucas numbers, it is easy to see that for a CA device under rule 184 with  $n$  cells:

- The number of states with no blocked car (“maximum speed” configurations) is  $L_n$  (example: for  $n = 4$  we have seven such configurations: 1010, 0101, 1000, 0100, 0010, 0001, 0000).
- The number of states with exactly one blocked car is  $nF_{n-2}$  (example: for  $n = 5$ , we have ten such configurations: 11000, 01100, 00110, 00011, 10001, 11010, 01101, 10110, 01011, 10101).

As a consequence, for such a CA model of traffic flow, the probability that a random distribution of vehicles “flowing freely” (i.e., no car is blocked) is

$$L_n/2^n \approx (\alpha/2)^n.$$

In general, one can prove the following generating function relation for the numbers  $v(n, k, l)$  of CA states in an  $n$ -cell device evolving under Rule 184, with  $k$  cars (1’s) such that  $l$  of them are able to move under the 184 updating process:

$$\left( \frac{q+1+\sqrt{(q-1)^2+4qz}}{2} \right)^n + \left( \frac{q+1-\sqrt{(q-1)^2+4qz}}{2} \right)^n = \sum_{k,l} v(n, k, l) q^k z^l.$$

Note that we may use the quantity  $l/k$  as a measure of the “average speed” in traffic (that is, the proportion of cars able to move).

Here is a list of selected MATLAB programs used in the aforementioned capstone project, instrumental in obtaining visuals on the density–speed relation and the traffic flow dynamics on an  $n$ -cell model, the intent being to witness, especially

near the critical point where the overall car density is given by  $k/n \approx 1/2$ , cluster formation or shock wave propagation (going forward or backward depending on the vehicle density in a given area). Note that such issues, especially in the “thermodynamic limit” (large lattice size) indicate bridges between traffic flow theory and statistical physics (Fuk s and Boccara 1998; Fukui and Ishibasi 1996).

**ardm(p)** This function generates a 0 or a 1 with the probability of a 1 being  $p$ .

```
function ardm=ardm(p)
x=rand(1);
if(x<p)
ardm=1;
else
ardm=0;
end
```

**cell(a,b,c)** This function represents the local neighborhood iteration according to Rule 184.

```
function cell = cell(a,b,c)
if 4*a+2*b+c==0
cell=0;
elseif 4*a+2*b+c==1;
cell=0;
elseif 4*a+2*b+c==2;
cell=0;
elseif 4*a+2*b+c==3;
cell=1;
elseif 4*a+2*b+c==4;
cell=1;
elseif 4*a+2*b+c==5;
cell=1;
elseif 4*a+2*b+c==6;
cell=0;
else
cell=1;
end
```

**next(x)** This function receives as input the bit vector  $x$  and produces a vector by applying Rule 184 to every local neighborhood of three cells.

```
function next = next(x)
n=size(x,2);
a(1)=cell(x(n),x(1),x(2));
a(n)=cell(x(n-1),x(n),x(1));
for I=2:n-1;
```

```

    a(I)=cell(x(I-1),x(I),x(I+1));
end;
next=a;

```

**nextflow(x)** This function receives as input a certain highway state  $x$  and calculates the average highway speed as the quotient of the number of vehicles that are actually moving and the total number of vehicles on the highway.

```

function nextflow = nextflow(x)
n=size(x,2);
d=0;
for I=1:n;
    d=d+x(I);
end;
for I=1:n-1;
    b(I)=x(I)-(x(I)*x(I+1));
    b(n)=x(n)-x(n)*x(1);
end;
c=0;
for I=1:n;
    c=c+b(I);
end;
nextflow=c/max(d,1);

```

**randvp(prob,n)** This function generates a random bit distribution on  $n$  cells, where 1's appear with the probability  $prob$ .

```

function randvp = randvp(prob,n)
for I=1:n;
    x(I)=randp(prob);
end;
randvp=x;

```

**spdensity(n)** This function produces a plot relating the car density and average car speed.

```

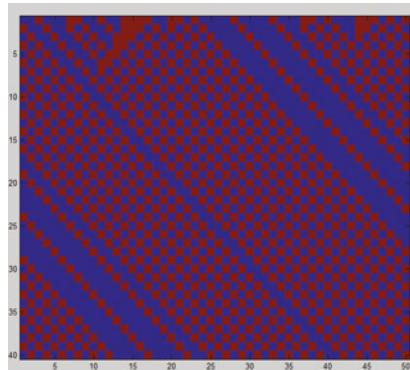
function spdensity = spdensity(n)
for I=1:n;
    x=randvp(I/n,n);
    y(I)=nextflow(x);
end;
spdensity=plot(y);

```

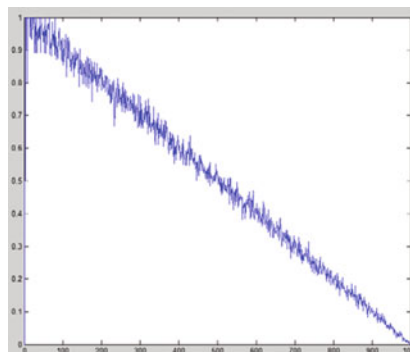
**rflow(n,c,k)** This function displays a visual of the traffic flow dynamics on an  $n$ -cell model, with an initial car distribution of density  $c$ , analyzed for the first  $k$  units of simulation time.

```
function rflow = rflow(n,c,k)
x=randvp(c,n);
a(1,:)=x;
for I=2:k;
    a(I,:)=next(a(I-1,:));
end
flow=imagesc(a);
```

For example, the following image is the output of **rflow(50, 0.4, 40)**, that is, we begin with a random distribution of “vehicles” (1’s in the automaton) with projected density 0.4, and run it for 40 steps. Hints of backward-moving waves can be noticed at the beginning (0.4 is not too far from the critical point 0.5), after which the movement becomes uniform, with each vehicle having an available empty cell to the right:



The following is the output of **spdensity(1000)**, illustrating through simulation the linear negative correlation between average vehicle speed and vehicle density:



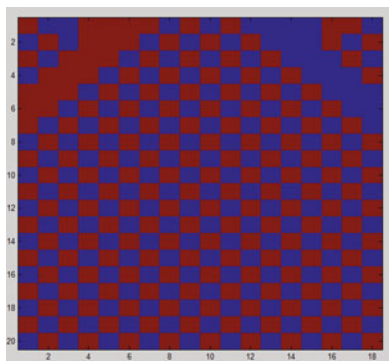
Inspired by the above traffic flow simulations, we find it exciting to try to apply the CA rule 184 to an initial structure consisting of a bit string of length  $p - 1$  (where  $p$  is a prime) with periodic boundary conditions, where the 1's indicate the locations of squares mod  $p$ , while the zeros indicate the locations of the nonsquares. In other words, the “vehicles” are the quadratic residues mod  $p$ . The following function creates a binary sequence of length  $p - 1$  obtained from the  $\pm 1$  sequence of Legendre symbols modulo  $p$  by replacing the  $-1$  values with 0.

```
function quadseq = quadseq(p)
for I = 1:p-1;
    x(I)=rem(I^2,p);
end
y=sort(x);
z=unique(y);
for I=1:p-1;
    s(I)=-1;
end
k=(p-1)/2;
for I=1:k
    s(z(I))=1;
end;
quadseq=s;
```

If we take this to be the initial “vehicle distribution” under the traffic CA rule, we will use the following function:

```
function quadflow = quadflow(p,k)
x=quadseq(p);
a(1,:)=x;
for I=2:k;
    a(I,:)=next(a(I-1,:));
end
quadflow=imagesc(a);
```

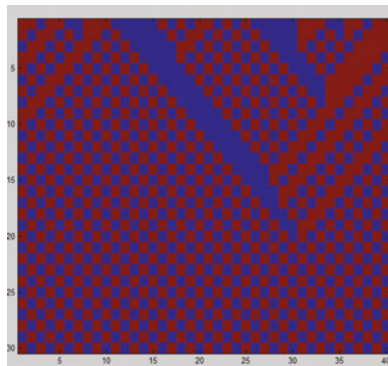
Let’s visualize what happens with, say, the small prime 19 if we perform 20 iterations:





Note that after an initial turbulence with a compact small group of vehicles briefly “flowing backward,” the circulating residues attain an equilibrium: a circular arrangement of 18 cells with periodic boundary conditions, in which every 1 is followed by a 0.

A similar phenomenon occurs if we try the prime 41. Again, the equilibrium is quickly attained after an initial traffic jam, with the 20 vehicle residues eventually traveling, equally spaced, at maximum speed:



An interesting question arises: is it true that if CA rule 184 is applied to the standard initial configuration of 1's and 0's corresponding to the quadratic residues and nonresidues modulo a prime, the configuration always stabilizes into one of equally spaced vehicles? For example, in the above scenario, the quadratic residues traffic stabilizes at the 20th step.

Here are some MATLAB functions used:

**npsum(x)** calculates the sum of the products of the nearest neighbors in a vector **x** (so that, particularly for the case of a binary “traffic vector” **x**, the value of **npsum(x)** is zero precisely when the traffic is free).

```
function [npsum]=npsum(x)
n=size(x,2);
for l=1:n-1;
    z(l)=x(l)*x(l+1);
end;
npsum=x(n)*x(1)+sum(z);
```

**quadflowend(p,k)** calculates the sums of the products of nearest neighbors for all subsequent vectors in the traffic iteration with  $k$  steps that starts with the binary vector of quadratic residues modulo  $p$  and places them in an output vector:

```
function quadflowend = quadflowend(p,k)
x=quadseq(p);
a(1,:)=x;
z(1)=npsum(x);
for l=2:k;
    a(l,:)=next(a(l-1,:));
    z(l)=npsum(a(l-1,:));
end;
quadflowend=z;
```

**testqfe(x)** calculates the level least  $k$  such that the components of  $x$  satisfy  $x_i = 0$  for all  $i \geq k$ :

```
function [testqfe] = testqfe(x)
testqfe=size(x,2);
l=testqfe
while x(l)~=0
    testqfe=testqfe-1;
    l=l-1;
end;
```

The following table indicates the level of stabilization for the primes from 7 to 100.

$p$	$L$	$p$	$L$
7	2	47	18
11	5	53	26
13	6	59	29
17	7	61	27
19	7	67	26
23	9	71	20
29	13	73	36
31	10	79	26
37	18	83	41
41	20	89	44
43	17	97	48

For an upper bound, it appears that a uniform flow is attained in fewer than  $\frac{p-1}{2}$  iterations. Here is a table with the number of iterations to stabilization for the segment of primes between 809 and 997:

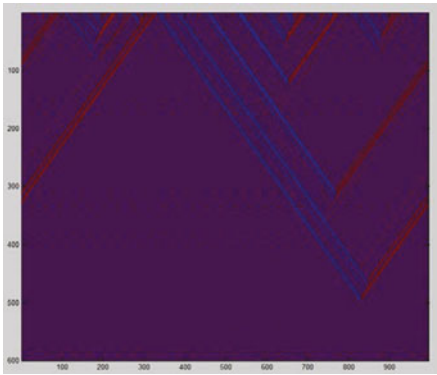
p	L	p	L
809	401	907	362
811	322	911	260
821	385	919	306

(continued)

(continued)

p	L	p	L
823	290	929	463
827	379	937	468
829	411	941	470
839	381	947	437
853	426	953	465
857	427	967	293
859	325	971	485
863	339	977	418
877	438	983	393
881	440	991	330
883	353	997	498
887	354		

The next image displays 600 iterations of the traffic map starting with the bit string corresponding to the quadratic residues and nonresidues modulo 997:



There is also the question of finding a lower bound for the number of iterations needed to stabilize the traffic. We can relate this to the sequence OEIS A048280 (the length of the longest run of consecutive quadratic residues modulo the  $n$ th prime) by noticing that a run of  $K$  consecutive 1's would need (if the environment allows, so at best)  $K - 1$  traffic-rule iterations in order to separate the 1's. Thus the largest residue run minus one makes for an obvious lower bound. Of course, there is still plenty of large scale computational work to do in this direction: as in all examples in this book, the reader is invited to join in the fun.

In the “further work” category, note that we can conceivably imagine plenty of other potentially interesting iterations starting from the binary quadratic residue seed. For example, imagine we want to apply a series of Ducci iterations. This time, the iteration does not preserve (as does CA rule 185) the number of 1's. Exploring the sequence of the periods of the binary Ducci iterations starting with these particular seeds may be an interesting task.

Since we will be working with binary vectors only, we will use the following MAPLE procedure for the Ducci iteration step:

```
> Ducci:= proc(x::list)::list;
> local n, k, X;
> n:=numelems(x);
> for k from 1 to n-1 do X[k]:= (x[k]+x[k+1]) mod 2 end do;
> X[n]:= (x[n]+x[1]) mod 2;
> [seq(X[k],k=1..n)];
> end proc;
```

When it comes to finding the limit cycle lengths, we will use the tortoise and the hare (Floyd's) cycle-finding approach outlined in Section 2.4. The Ducci period corresponding to the quadratic residue seed associated to the prime  $p$ , say  $p = 103$ , is 510, and it can be obtained as follows:

```
p:=103: for k from 1 to p-1 do s(k):=(1+legendre(k,p))/2:
end do: L:= [seq(s(k),k=1..p-1)]: SLOW:=Ducci(L):
FAST:=Ducci(Ducci(L)): k:=1: while SLOW<>FAST do
SLOW:=Ducci(SLOW): FAST:=Ducci(Ducci(FAST)): k:=k+1: end do:
TENTATIVE_START:=k: x:=SLOW: SLOW:=Ducci(x):
FAST:=Ducci(Ducci(x)): k:=1: while SLOW<>FAST do
SLOW:=Ducci(SLOW): FAST:=Ducci(Ducci(FAST)): k:=k+1: end do:
PERIOD:=k;
```

The calculations of Ducci periods thus defined vary greatly in both output value and the running time of the cycle-finding algorithm (which is conceivable if we think about the significant differences between the multiplicative structures of the numbers  $p - 1$ . While a pattern is not clearly detectable to us, we include a table with a couple of the periods associated to the primes from 3 to 103; the largest entry, 950,214, corresponds to the prime 59.

$p$	<i>period</i>	$p$	<i>period</i>
3	1	47	4094
5	1	53	3276
7	6	59	950,214
11	30	61	60
13	12	67	2046
17	1	71	8190
19	126	73	504
23	682	79	8190
29	28	83	83,886
31	30	89	2728
37	252	97	96
41	120	101	102,300
43	126	103	510

As an exercise, we suggest the following computational project:

**Computational Exploration Project CEP 1** Find the period of the 106-number Ducci game played starting with the binary seed [1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1]

indicating the quadratic residues (1's) and nonresidues (0's) in  $F_{107}^*$ .

<http://www.springer.com/978-3-319-56761-7>

Sequential Experiments with Primes

Caragiu, M.

2017, XI, 279 p. 89 illus., 50 illus. in color., Hardcover

ISBN: 978-3-319-56761-7