Lecture 13: Experimental mathematics

1. Objective

We look at some problems in which computers could play a role. Just if a counter example would exist:

2. Goldbach's conjecture

The statement is that every even number larger than 2 is the sum of two primes. While it is unlikely that a computer search will find a counter example, it might be that computer search finds the solution to the problem. How?

3. Euler Cuboid

It is not known whether there exists a cuboid with integer side length such that all face diagonals are integers and additionally, also the large diagonal is an integer. Computer searches are on the way. One can search on parametrized families of Euler cubes.

4. Riemann hypothesis

This is a prototype where experimens led to more and more support that the Riemann hypothess is true. One approach looks numerically for roots of the Riemann zeta function on the critical line. An other approach is the Mertens approach. Define $\mu(n) = 1$ if n is the product of an even number of different primes and -1 if n is the product of an odd number of different primes. In all other cases, that is if n has a square factor larger than 1, we have $\mu(n) = 0$. Is μ sufficiently random so that $S_n = \sum_{k=1}^n \mu(k)$ grows like the iterated law of logarithm? While it looks as if the $\mu(n)$ behave like a random sequence, there are some correlations.

5. Blliards

One does not know whether there are triangular billiards without periodic points.
One also does not know whether there are smooth convex billiards besides the ellipse for which one has integrability in the sense that all points are either periodic, asymptotic to a periodic point or almost periodic.

6. Chaos in the standard map

We have seen the problem before. Verify that for $c > 2$ and all n $\frac{1}{n}\int_0^1\int_0^1 \log|\partial_x f_n(x,y)|\, dxdy \geq \log(\frac{c}{2})$. The left hand side converges to the average of the Lyapunov exponents which is in this case also the **entropy** of the map.

5. Are there prime twins?

This is a problem, which can first of all be investigated statistically. Similarly as Gauss looked numerically for a law describing the frequency of the prime numbers, one can first see, hoe many prime twins one has to expect in a certain interval and then see whether this expectation is confirmed. Furthermore one can look whether there are any patterns on arithmetic subsequences. Maybe there are some unexpected sequences along which there are more prime twins. Related to the twine prims problem is the problem to estimate the minimal distance between two Gaussian primes in the complex plane.

5. Is π normal?

The decimal digits of π appear random enough so that every digit appears with the same frequency.

6. Are there odd perfect numbers?

While a brute force search is unlikely, there are other approaches which are more likely to find an odd perfect number. Any perfect number satisfies $\sigma(n)/n = 2$, where $\sigma(n)$ is the sum of all the divisors of n including n. Take a large set $B = \{p_1, ..., p_s\}$ of primes. For every $k = (k_1, ..., k_s)$, form the number $n = p_1^{k_1} p_2^{k_2} ... p_s^{k_l}$. We have $\sigma(p^k) = (1 + p + p^2 + ...p^k)/p^k = (p - p^{-k})/(p - 1)$. Let $a(p, k) = \log(\sigma(p^k))$. The goal is to find $(k_1, ..., k_s)$ such that

$$\log(a(p_1, k_1)) + ... + \log(a(p_l, k_l)) - k_1 \log(p_1) - ...k_s \log(p_s) = \log(2) \; .$$

The idea is to keep first the primes and change only the "dials" k_j in a controlled way. Certain dial changes will produce a very small net change of the left hand side. For large primes and large n the first order change will dominate and methods from Diophantine geometry could be used and linear algebra to get close to the right hand side. If lucky (provided of course there is an odd perfect number), one could hit it like this. If we would take 1000 primes each 1000 digits long and deal with exponents of the order of 1000, we would investigate numbers with billions of digits.

Worksheet: Turing machines

1. Objective

We see a concrete Turing machine and evolve it a few steps to see what it does.

2. The machine

The machine is initially in state [1] and starts with an empty tape.

	-3	-2	-1	[0]	1	2	3	···
···	**0**	**0**	**0**	**0**	**0**	**0**	**0**	···

Here is the definition of a machine with three states:

Input 0

State	New state	Write	move to
[1]	[2]	0	left
[2]	[3]	1	right
[3]	[1]	1	left

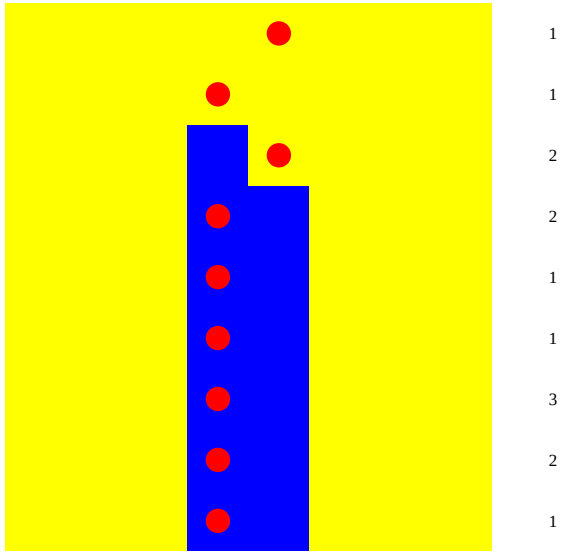
Input 1

State	New state	Write	move to
[1]	[2]	1	left
[2]	[1]	0	right
[3]	[3]	0	right

2. Work the machine

We run this machine a few steps and mark the number, where the machine reads the tape. This place will move with time.

	-3	-2	-1	[0]	1	2	3	···	State=1
···	**0**	**0**	**0**	**0**	**0**	**0**	**0**	···	State=1
···	**0**	**0**	**1**	**0**	**0**	**0**	**0**	···	State=2
···	**0**	**0**	**1**	**1**	**0**	**0**	**0**	···	State=3
···	**0**	**0**	**1**	**1**	**0**	**0**	**0**	···	State=1
···	**0**	**0**	**1**	**1**	**0**	**0**	**0**	···	State=1



3. A new machine

Here is the definition of a new machine with three states:

Input 0

State	New state	Write	move to
[1]	[2]	1	right
[2]	[3]	1	right
[3]	[3]	0	left

Input 1

State	New state	Write	move to
[1]	[2]	1	left
[2]	[1]	0	right
[3]	[3]	0	left

2. It's your turn!

Run it!

···	**0**	**0**	**0**	**0**	**0**	**0**	**0**	···	State=1
···								···	State=
···								···	State=
···								···	State=

Lecture 13: worksheet the integer partition problem

1. Objective

We look at a problem which is known to be **NP-complete**. If it could be solved in polynomial time, all NP problems would be polynomial and P=NP.
In other words, if you can design a method, which solves the problem in a manner which is polynomial in n, you win a Million dollars and you would have solved the most important problem in computer science.

2. The problem

Given n positive integers $a_1, ..., a_n$, divide them up into two subsets, so that the sum of these numbers in one set and the sum of numbers in the other set are as close together as possible.

3. An example

Assume

$$A = \{2, 3, 5, 7, 11, 13, 17\} .$$

1. Apply the "cruel school selection" algorithm. The largest two elements are the team leaders of a team, who alternatively select the best remaining candidate. What do you get?

2. Can you do better than that? How well can you divide up the numbers better?

Lecture 13: Worksheet factoring integers

1. Objective

One does not know how to factor integers of 200 digits efficiently. Can we find some factors fast?

2. Checking a factor of 9

1) There is a simple method to decide whether a number is divisible by 9. What is it?

2) For example: is

$$121212121212121212$$

divisible by 9?

3) There is a simple method to decide whether a number is divisible by 11. What is it?

4) For example is

$$12121212121212121212$$

divisible by 11?

5) Which factors can be detected by looking at the last decimal digit of a number only?

6) The 1,-3,2 method for divisibility by 7 is to split the number into groups of 6, then multiply in each group the three pairs by 1,-3,2, then add up all results. For example $11111111111114 = (000011)(111111)(111114)$ leads to $2*11+1*11-3*11+2*11+1*11-3*11+2*14 = 28$ which is divisible by 7.

Question: is

$$111111111111111111111111111111111111111111111111$$

divisible by 7 (48 digits of 1).