Bachelor of Ecole Polytechnique
Algorithms for Discrete Mathematics, year 2, semester 1

Experimental Maths 2: Matrices, Modulos and Fermat

Table of contents


```
# execute this part to modify the css style
from IPython.core.display import HTML
def css_styling():
    styles = open("./style/custom2.css").read()
    return HTML(styles)
css_styling()
```

```
## loading python libraries

# necessary to display plots inline:
%matplotlib inline

# load the libraries
import matplotlib.pyplot as plt # 2D plotting library
import numpy as np              # package for scientific computing
from math import *              # package for mathematics (pi, arctan, sqrt, factorial
```

Arithmetic with matrices

The aim of this Section is to use linear algebra and python to compute *exact* expressions in Arithmetic.

Do it yourself. Theory

1. Prove by induction that there exist integers a_n, b_n such that for every $n \geq 1$
$$(1+\sqrt{2})^n = a_n + b_n\sqrt{2}.$$
2. Find a 2×2 matrix A such that

$$\begin{pmatrix} a_{n+1} \\ b_{n+1} \end{pmatrix} = A \times \begin{pmatrix} a_n \\ b_n \end{pmatrix}.$$

Answers.

1.
2.

Do it yourself. Using the powers of matrix A, write a small script which computes the exact value of $(1+\sqrt{2})^{100}$.

Do it yourself.

We set

$$u_1 = \frac{1}{1+1}, \quad u_2 = \frac{1}{1+\frac{1}{1+1}}, \quad u_3 = \frac{1}{1+\frac{1}{1+\frac{1}{1+1}}}, \quad u_4 = \frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{1+1}}}}, \ldots$$

1. **(Theory)** Let us write u_n as an integer ratio $u_n = a_n/b_n$ (for example $a_1 = 1, b_1 = 2$, you can check $a_3 = 3, b_3 = 5$). Find a 2×2 matrix B such that
$$\begin{pmatrix} a_{n+1} \\ b_{n+1} \end{pmatrix} = B \times \begin{pmatrix} a_n \\ b_n \end{pmatrix}.$$
(Hint: Find a simple relation between u_{n+1} and u_n.)
2. **(Application)** Compute the exact value (as a fraction) of
$$r = \frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{1+1}}}}}}}}}.$$
3. **(Side question)** Compute a numerical evaluation of $1+r$. Does it look familiar?

Answers.

1.
2.
3.

Do it yourself. Theory

1. Find a function f such that for every $n \geq 0$
$$u_{n+1} = f(u_n).$$

2. Compute the limit of (u_n) (assuming the limit exists). Compare with your numerical result above.

Answers.

1.
2.

Arithmetic of modulos

Do it yourself. Write a script which computes $38911^{21025413} \pmod{188}$. (Explain in the cell below the successive steps.)

Answers.

The little Fermat theorem

The "little" Fermat Theorem states the following:

Theorem
Let p be a prime number. For every integer $1 \leq a < p$, we have

$$a^{p-1} \equiv 1 (\text{mod } p).$$

(In the sequel we will use the above formulation rather than "for every $a \geq 0$, we have $a^p \equiv a (\text{mod } p)$".)

Do it yourself. Write a script which checks that the little Fermat Theorem is true for $p = 17$.

We say that n is *composite* if n is not prime. The contraposition of the little Fermat Theorem is very useful: it says that

$$\left(\text{there exists } a < p \text{ such that } a^{p-1} \not\equiv 1 \pmod{n}\right) \Rightarrow p \text{ is composite.}$$

In this case, we say that a is a *Fermat witness* for (the non-primeness of) p. For example, you can check that

$$2^{15-1} = 16384 \equiv 4 \pmod{15} \not\equiv 1 (\text{mod } 15)$$

(and of course 15 is composite). Therefore $a = 2$ is a Fermat witness for $p = 15$, and this is a (somehow convoluted) proof of the fact that 15 is composite.

Do it yourself.

1. Check that for every composite $n \leq 60$ then $a = 2$ is a Fermat witness for n. The output should look like

```
n = 2 is prime
n = 3 is prime
n = 4 is composite and 2 is a Fermat witness
n = 5 is prime
n = 6 is composite and 2 is a Fermat witness
...
```

2. Find the smallest composite n such that $a = 2$ is not a Fermat witness for n.
3. Same question with $a = 3$.

To save you time we have copy/pasted the function `IsPrime()` from Notebook 1:

```
def IsPrime(n):
    # input: integer n
    # output: True or False depending on whether n is prime or not
    if n==1:
        return False
    if n==2:
        return True
    elif n%2==0:
        return False
    factor=3
    while factor**2 < n+1:
        if n%factor == 0:
            return False
        factor=factor+2
    return True

# Tests
print(IsPrime(2))
print(IsPrime(108))
```

```
# Question 1
print('----------Question 1-----------')

# Question 2

print('')
print('----------Questions 2-3-----------')
```

Do it yourself. Find the smallest Fermat witness which proves that 1105 is not prime.

Fermat prime numbers

A *Fermat number* is an integer of the form $F_n = 2^{2^n} + 1$ for some $n \geq 0$. First Fermat numbers are given by

$$F_0 = 2^1 + 1 = 3, \quad F_1 = 2^2 + 1 = 5, \quad F_2 = 2^4 + 1 = 17, \quad , F_3 = 2^8 + 1 = 257.$$

> **Do it yourself.** Fermat conjectured that every Fermat number is prime. Can you test his conjecture up to $n = 8$?