Bachelor of Ecole Polytechnique
Computational Mathematics, year 2, semester 1

Experimental Mathematics 3: Test

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
```

1. Arithmetic with matrices

Do it yourself.
(Theory) Let i be the complex number $i^2 = -1$ and for $n \geq 1$ let a_n, b_n be the integers such that

$$(1+5i)^n = a_n + ib_n.$$

(You don't need to prove that these integers exist.)

1. 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.

Do it yourself. Using the powers of matrix A, write a small script which computes $(1+5i)^{100}$.

2. Sums of two squares

We say that n is a sum of two squares if there exist two integers $a, b \geq 1$ such that
$$n = a^2 + b^2.$$
For example, $10 = 3^2 + 1^2$ is a sum of two squares while 11 is not.

Do it yourself.
Write a function `SumsOfTwoSquares(n)` which returns the list of all decompositions of n as a sum of two squares. For example:

```
SumsOfTwoSquares(905)
[[8, 29], [11, 28], [28, 11], [29, 8]]
SumsOfTwoSquares(11)
[]
```

Do it yourself. Find the smallest integer n which has strictly more than 7 decompositions as a sum of two squares.

Do it yourself.

1. **(Theory)** Let a be any integer. Prove that a^2 is always equal to 0 or 1 modulo 4.
2. **(Theory)** Deduce that $\left(n \equiv 3 \pmod 4\right) \Rightarrow \left(n \text{ is not a sum of two squares}\right)$.
3. Is the converse true?

Answers.

1.

2.
3.

3. Multiplicative persistence

Do it yourself.
Write a function `ProductOfDigits(n)` which returns the product of digits of n. For example, `ProductOfDigits(2281)=32` , since $2 \times 2 \times 8 \times 1 = 32$.

For an integer n, we consider the following procedure. Multiply all the digits of n by each other, repeating with the product until a single digit is obtained. The number of steps required is called the *multiplicative persistence* of n, and is denoted by $\mathrm{Mp}(n)$. (If $n < 10$ we set $\mathrm{Mp}(n) = 0$.)

For example, for $n = 2281$:

$$2281 \overset{\text{1st step}}{\longrightarrow} 2 \times 2 \times 8 \times 1 = 32 \overset{\text{2d step}}{\longrightarrow} 3 \times 2 = 6.$$

Therefore $\mathrm{Mp}(2281) = 2$.

Do it yourself. Write a function `MultiplicativePersistence()` which computes the multiplicative persistence.

Do it yourself. Find the smallest integer K which has a multiplicative persistence larger (or equal) to 7.

Do it yourself.
1. Let $\Pi(n)$ denote the product of digits of n. Prove that $\Pi(n) < n$ for every $n \geq 10$.
2. Deduce that the multiplicative persistence of an integer is always finite.
3. **(Bonus)** Can you find a good upper bound for $\mathrm{Mp}(n)$ as a function of n?

Answers.
1.
2.
3.