Bachelor of Ecole Polytechnique
Computational Mathematics, year 2, semester 1
Lecturer: Lucas Gerin (send mail) (mailto:lucas.gerin@polytechnique.edu

Symbolic computing 2: Generating functions with SymPy

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 pylab import *

from math import *              # package for mathematics (pi, arctan, sqrt, factori
import sympy as sympy             # package for symbolic computation
from sympy import *
```

Basics of generating functions

Let us first explain how we will manipulate generating functions with `SymPy`. We consider the example of

$$f(x) = \frac{1}{1-2x} = 1 + 2x + 4x^2 + 8x^3 + 16x^4 + \dots$$

We first introduce a symbolic variable x and an expression f as follows:

```
x=var('x')
f=(1/(1-2*x))

print('f = '+str(f))
print('-----')
print('We check that coefficients are correct:')
print('series expansion of f at 0 and of order 10 is: '+str(f.series(x,0,10)))
```

```
f = 1/(-2*x + 1)
-----
We check that coefficients are correct:
series expansion of f at 0 and of order 10 is: 1 + 2*x + 4*x**2 + 8*x**
3 + 16*x**4 + 32*x**5 + 64*x**6 + 128*x**7 + 256*x**8 + 512*x**9 + O(x*
*10)
```

One can extract n-th coefficient as follows:

- f has to be truncated at order k (for some $k > n$) with `f.series(x,0,k)`
- the n-th coefficient is then extracted by `f.coeff(x**n)`

```
f_truncated = f.series(x,0,8)
print('Truncation of f is '+str(f_truncated))
n=6
nthcoefficient=f_truncated.coeff(x**n)
print(str(n)+'th coefficient is: '+str(nthcoefficient))
```

```
Truncation of f is 1 + 2*x + 4*x**2 + 8*x**3 + 16*x**4 + 32*x**5 + 64*x
**6 + 128*x**7 + O(x**8)
6th coefficient is: 64
```

Exercise 1. Fibonacci generating function

In class we proved that the generating function of the Fibonacci sequence is given by:

$$\mathcal{F}(x) = \frac{1}{1 - x - x^2}.$$

> Do it yourself.
>
> 1. Write a recursive function `Fibonacci(n)` which returns the n-th Fibonacci number.
> 2. Write another function `FibonacciGF(n)` which also returns the n-th Fibonacci number by extracting the n-th coefficient in $\mathcal{F}(x)$.

```
def Fibonacci(n):
    if n==0 or n==1:
        return 1
    else:
        return Fibonacci(n-1)+Fibonacci(n-2)

def FibonacciGF(n):
    x=var('x')
    f=(1/(1-x-x**2))
    f_truncated=f.series(x,0,n+1)
    return f_truncated.coeff(x**n)

print([Fibonacci(n) for n in range(1,20)])
print([FibonacciGF(n) for n in range(1,20)])
```

```
[1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584
, 4181, 6765]
[1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584
, 4181, 6765]
```

Do it yourself.

1. For both algorithms, plot the execution time of `Fibonacci(n)` and `FibonacciGF(n)` (say, for $1 \leq n \leq 35$).

   Use the library `time`. For instance the following script returns the execution time of `Fibonacci(12)`:

   ```
   import time
   time1=time.clock()
   Fibonacci(12)
   time2=time.clock()
   print(time2-time1)
   ```

2. What do you observe?

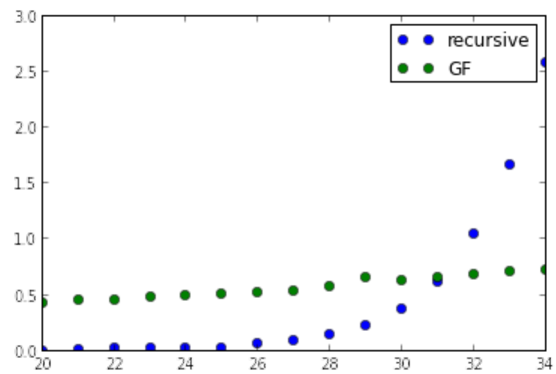
```python
import time

RunningTimeFibonacci=[]
RunningTimeFibonacciGF=[]
a=20
b=35

for n in range(a,b):
    time1=time.clock()
    Fibonacci(n)
    time2=time.clock()
    RunningTimeFibonacci.append(time2-time1)

for n in range(a,b):
    time1=time.clock()
    FibonacciGF(n)
    time2=time.clock()
    RunningTimeFibonacciGF.append(time2-time1)

N=range(a,b)
plt.plot(N,RunningTimeFibonacci,'o',label='recursive')
plt.plot(N,RunningTimeFibonacciGF,'o',label='GF')
plt.legend()
plt.show()
```



> Answers. The running time of the GF-algorithm seems linear while the running time of the recursive functions looks exponential. For $n \geq 30$ the GF-algorithm is faster.

Exercise 2. Recurrence of order two and asymptotics

Let j_n be defined by

$$\begin{aligned} j_0 &= 0, \\ j_1 &= 1, \\ j_2 &= 2, \\ j_n &= 2j_{n-2} + 5 \qquad (\text{ for every } n \geq 3). \qquad (\#) \end{aligned}$$

> Do it yourself. (Theory) Find the expression for the generating function $J(x)$ of the j_n's.
> *(Recall that you can ask* `SymPy` *to solve any equation.)*

Answers.

1. We multiply eq. (#) by x^n and sum the resulting expression for all $n \geq 3$:

$$\sum_{n\geq 3} j_n x^n = 2\sum_{n\geq 3} j_{n-2}x^n + 5\sum_{n\geq 3} x^n$$

$$J(x) - j_1 x - j_2 x^2 = 2\sum_{p\geq 1} j_p x^{p+2} + 5(x^3 + x^4 + x^5 + \dots) \qquad \text{(we put } n-2=p\text{)}$$

$$J(x) - x - 2x^2 = 2x^2 J(x) + 5\frac{x^3}{1-x}.$$

We solve this last equation with the following script:

```
x=symbols('x')
j=symbols('j')
SeriesJ=solve(j-x-2*x**2-2*x**2*j-5*x**3/(1-x),j)
print('The solution is: J(x)= '+str(latex(SeriesJ[0])))
```

```
The solution is: J(x)= \frac{x \left(3 x^{2} + x + 1\right)}{2 x^{3} -
2 x^{2} - x + 1}
```

Answers. We find

$$J(x) = \frac{x\left(3x^2 + x + 1\right)}{2x^3 - 2x^2 - x + 1}$$

Do it yourself.

1. Write a function which extracts the n-th coefficient in $J(x)$.
2. Compare your results with a recursive function which computes the j_n's.

```python
# Question 1
def j_GF(n):
    x=symbols('x')
    j= Function('j')
    j=((x*(3*x**2+x+1)/(1-x-2*x**2+2*x**3)))
    j_truncated=j.series(x,0,n+1)
    return j_truncated.coeff(x**n)


print('With GFs we obtain:')
print([j_GF(n) for n in range(1,15)])

# Question 2

def j_recursive(n):
    if n==1:
        return 1
    elif n==2:
        return 2
    else:
        return 2*j_recursive(n-2)+5

print('With recursion we obtain:')
print([j_recursive(n) for n in range(1,15)])
```

```
With GFs we obtain:
[1, 2, 7, 9, 19, 23, 43, 51, 91, 107, 187, 219, 379, 443]
With recursion we obtain:
[1, 2, 7, 9, 19, 23, 43, 51, 91, 107, 187, 219, 379, 443]
```

Do it yourself.

1. What is the radius of convergence of $J(x)$? (You can ask help to SymPy.)
2. What does it imply for the asymptotic behaviour of j_n? (Apply the "exponential growth formula", that we saw in class.)

```python
solve(2*x**3-2*x**2-x+1,x)
```

```
[1, -sqrt(2)/2, sqrt(2)/2]
```

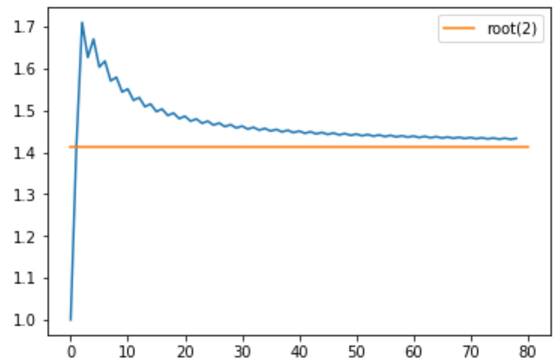
Answers.

1. The radius of convergence ρ of $\mathcal{J}(x)$ is such that $2\rho^3 - 2\rho^2 - \rho + 1 = 0$, *i.e.*
$$\rho = \sqrt{2}/2 = 1/\sqrt{2}$$
according to the previous script.
2. The exponential growth formula says that for every ε there are $C_1, C_2 > 0$ such that and for n large enough
$$C_1(\sqrt{2} - \varepsilon)^n = C_1(1/\rho - \varepsilon)^n \leq J_n \leq C_2(1/\rho + \varepsilon)^n = C_2(\sqrt{2} + \varepsilon)^n.$$
(The exponential growth formula only ensures that the left inequality holds for infinitely many n's.)

Do it yourself. With a plot, find an approximation of r such that j_n grows like $\mathrm{const} \times r^n$. Compare with the previous question.

```
N=80
RenormalizedValues=[j_recursive(n)**(1/(n+0.0)) for n in range(1,N)]
plt.plot(RenormalizedValues)
plt.plot([0,N],[np.sqrt(2),np.sqrt(2)],label='root(2)')

plt.legend()
plt.show()
```



Answers. We plot $n \mapsto (j_n)^{1/n}$ which seems to converge to $\sqrt{2}$. This is consistent with the exponential growth formula.

Exercise 3. A pair of generating functions

Let a_n, b_n be defined by $a_1 = b_1 = 1$ and, for every $n \geq 1$,

$$\begin{cases} a_{n+1} &= a_n + 2b_n, \\ b_{n+1} &= a_n + b_n. \end{cases} \qquad (\&)$$

(This is our example of Notebooks 2 and 4: a_n, b_n are defined by $(1+\sqrt{2})^n = a_n + b_n\sqrt{2}$.

Do it yourself.

1. Find a 2×2 system whose solutions are $A(x), B(x)$, where A, B are the generating functions of sequences $(a_n)_{n\geq 1}, (b_n)_{n\geq 1}$. *(Coefficients of this system should depend on x.)*
2. Solve this system with `solve` and write a script which uses function A to return $a_1, \dots, a_{20}$.

Answers.

1. We multiply both terms of both equations in eq.(&) by x^{n+1} and take the sum for $n\geq 1$. We obtain

$$\begin{cases} \sum_{n\geq 1} a_{n+1}x^{n+1} &= \sum_{n\geq 1} a_n x^{n+1} + 2\sum_{n\geq 1} b_n x^{n+1}, \\ \sum_{n\geq 1} b_{n+1}x^{n+1} &= \sum_{n\geq 1} a_n x^{n+1} + \sum_{n\geq 1} b_n x^{n+1}. \end{cases}$$

We find

$$\begin{cases} A(x)-x &= xA(x)+2xB(x), \\ B(x)-x &= xA(x)+xB(x). \end{cases}$$

In the above script we solve this sytem of equations. We find

$$\begin{cases} A(x) &= -\frac{x(x+1)}{x^2+2x-1} \\ B(x) &= -\frac{x}{x^2+2x-1} \end{cases}.$$

```
#---- Question 2
# We solve the system
var('A B x')
Solutions=solve([A-x-x*A-2*x*B,B-x-x*A-x*B],[A,B])
print(Solutions)
# We obtain the following expression:
FunctionA=-x*(x + 1)/(x**2 + 2*x - 1)
FunctionB=-x/(x**2 + 2*x - 1)
print("A(x) = "+str(FunctionA.series(x,0,10)))
print("B(x) = "+str(FunctionB.series(x,0,10)))

# We extract a_1,a_2,...
N=20

A_truncated = FunctionA.series(x,0,N+2)
FirstCoefficients= [A_truncated.coeff(x**n) for n in range(1,N+1)]
print(FirstCoefficients)
```

```
{B: -x/(x**2 + 2*x - 1), A: -x*(x + 1)/(x**2 + 2*x - 1)}
A(x) = x + 3*x**2 + 7*x**3 + 17*x**4 + 41*x**5 + 99*x**6 + 239*x**7 + 5
77*x**8 + 1393*x**9 + O(x**10)
B(x) = x + 2*x**2 + 5*x**3 + 12*x**4 + 29*x**5 + 70*x**6 + 169*x**7 + 4
08*x**8 + 985*x**9 + O(x**10)
[1, 3, 7, 17, 41, 99, 239, 577, 1393, 3363, 8119, 19601, 47321, 114243,
275807, 665857, 1607521, 3880899, 9369319, 22619537]
```

Automatic decomposition of fractions

In class we saw that for GFs it is useful to decompose fractions like this:

$$\frac{1-x+x^2}{(1-2x)(1-x)^2} = \frac{3}{1-2x} - \frac{1}{1-x} - \frac{1}{(1-x)^2}.$$

Here are examples on how to do that with SymPy.

Exercise 3 (continued)

Do it yourself. The goal of the exercise is to find coefficients α, β, a, b, c such that

$$A(x) = \frac{a}{x-\alpha} + \frac{b}{x-\beta} + c,$$

where

$$A(x) = \frac{-x\,(x+1)}{x^2+2x-1}$$

was defined in the previous exercise.

1. (Theory) Compute $\lim_{x\to+\infty} A(x)$ and deduce c.
2. (Theory + SymPy) Use `SymPy` to find coefficients α, β.
3. (Theory + SymPy) Use `SymPy` again to find coefficients a, b.

Answers. Question 1. Taking the limit $(x \to +\infty)$ in the equation

$$\frac{a}{x-\alpha} + \frac{b}{x-\beta} + c = \frac{-x\,(x+1)}{x^2+2x-1}$$

yields $0+0+c=-1$.

```
# Question 2
# We solve "denominator of A = 0" to find a,b
var('x')
solve(x**2+2*x-1,x)
```

```
[-1 + sqrt(2), -sqrt(2) - 1]
```

Answers. Question 2.
The short code above shows that

$$A(x) = \frac{-x\,(x+1)}{x^2+2x-1} = \frac{-x\,(x+1)}{(x-(\sqrt{2}-1))(x-(-\sqrt{2}-1))}$$

so we must have that $\alpha = \sqrt{2}-1$ and $\beta = -\sqrt{2}-1$ (or the contrary) to ensure that A has the proper definition domain.

```
# Question 3
# We use A(1), A(0) to find alpha,beta

var('a b alpha beta x')

alpha=-1 + sqrt(2)
beta=-sqrt(2) - 1
# Left-hand side:
def A_factorized(x):
    return -(x*(x+1))/(x**2+2*x-1)

# Right-hand side:
def A_decomposed(x,a,b,alpha,beta):
    return a/(x-alpha)+b/(x-beta)+(-1)

# we identify a,b,c by solving the following system:
Solutions=solve([A_factorized(0)-A_decomposed(0,a,b,alpha,beta),A_factorized(1)-A_de
print(Solutions)
astar=Solutions[a]
bstar=Solutions[b]

# to get a nice formula:
var('x')
print(latex(A_decomposed(x,astar,bstar,alpha,beta)))
```

```
{b: 1/2 + sqrt(2)/2, a: -sqrt(2)/2 + 1/2}
-1 + \frac{- \frac{\sqrt{2}}{2} + \frac{1}{2}}{x - \sqrt{2} + 1} + \fra
c{\frac{1}{2} + \frac{\sqrt{2}}{2}}{x + 1 + \sqrt{2}}
```

Answers. In order to find a, b we solve with the previous code the system

$$\begin{cases} A(0) & = \frac{a}{0-\alpha} + \frac{b}{0-\beta} + (-1) \\ A(1) & = \frac{a}{1-\alpha} + \frac{b}{1-\beta} + (-1) \end{cases}$$

with $\alpha = \sqrt{2} - 1$ and $\beta = -\sqrt{2} - 1$. We find `{b: 1/2 + sqrt(2)/2, a: -sqrt(2)/2 + 1/2}` and finally

$$A(x) = -1 + \frac{-\frac{\sqrt{2}}{2} + \frac{1}{2}}{x - \sqrt{2} + 1} + \frac{\frac{1}{2} + \frac{\sqrt{2}}{2}}{x + 1 + \sqrt{2}}.$$

Do it yourself. (Bonus: Theory) Deduce a proof of the formula

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

(we obtained this formula in the previous notebook).

(Hint: Use the formula

$$\frac{1}{x - \rho} = -\frac{1/\rho}{1 - x/\rho} = -1/\rho \sum_{n \geq 0} x^n (1/\rho)^n. \tag{E}$$

Answers. We have that

$$A(x) = -1 + \frac{-\frac{\sqrt{2}}{2} + \frac{1}{2}}{x-\alpha} + \frac{\frac{1}{2} + \frac{\sqrt{2}}{2}}{x-\beta}$$

with $\alpha = \sqrt{2} - 1$ and $\beta = -\sqrt{2} - 1$. Using equation (E) with $\rho = \alpha$ and $\rho = \beta$ we obtain

$$\begin{aligned} A(x) &= -1 + \frac{1-\sqrt{2}}{2}(-1/\alpha \sum_{n\geq 0} x^n (1/\alpha)^n) + \frac{1+\sqrt{2}}{2}(-1/\rho \sum_{n\geq 0} x^n (1/\beta)^n) \\ &= -1 + \frac{1}{2}\sum_{n\geq 0} x^n (1/\alpha)^n + \frac{1}{2}\sum_{n\geq 0} x^n (1/\beta)^n \\ &= -1 + \frac{1}{2}\sum_{n\geq 0} x^n (1+\sqrt{2})^n + \frac{1}{2}\sum_{n\geq 0} x^n (1-\sqrt{2})^n \end{aligned}$$

since $1/\alpha = 1 + \sqrt{2}$ and $1/\beta = 1 - \sqrt{2}$.

By identification of coefficients we obtain

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

Exercise 4: (Bonus) To practice: Another decomposition of fractions.

Do it yourself. Set

$$f(x) = \frac{1}{2x^3 - 10x^2 - 30x + 54}.$$

With `SymPy` find reals $a, b, c, \alpha, \beta, \gamma$ such that

$$f(x) = \frac{a}{x-\alpha} + \frac{b}{x-\beta} + \frac{c}{x-\gamma}.$$

(You can use `solve` *,* `simplify` *,* `expand` *. Remember to use* `Rational(,)` *for fractions.)*

Answers. First, we observe that $\frac{1}{2x^3-10x^2-30x+54}$ and $\frac{a}{x-\alpha} + \frac{b}{x-\beta} + \frac{c}{x-\gamma}$ must have the same domain of definition. Therefore

$$\{\text{Roots of } 2x^3 - 10x^2 - 30x + 54\} = \{\alpha, \beta, \gamma\}.$$

In the script below we obtain:

$$\alpha = -3, \beta = -\sqrt{7} + 4, \gamma = \sqrt{7} + 4.$$

```
var('x a b c')
Num=2*x**3 - 10*x**2 - 30*x + 54
Roots=solve(Num,x)
print(Roots)
[alpha, beta, gamma] = Roots
```

```
[-3, -sqrt(7) + 4, sqrt(7) + 4]
```

Answers. We must have

$$\frac{1}{2x^3 - 10x^2 - 30x + 54} = \frac{a}{x+3} + \frac{b}{x-4+\sqrt{7}} + \frac{c}{x-4-\sqrt{7}}.$$

for $x = 0, 1, 2$ We obtain with the following script that

$$\frac{-\frac{\sqrt{7}}{168} - \frac{1}{168}}{x-4+\sqrt{7}} + \frac{-\frac{1}{168} + \frac{\sqrt{7}}{168}}{x-4-\sqrt{7}} + \frac{\frac{1}{84}}{(x+3)}$$

```
# Left-hand side:
def LHS(x):
    return Rational(1,2*x**3 - 10*x**2 - 30*x + 54)

# Right-hand side:
def RHS(x,a,b,c):
    return a/(x-alpha)+b/(x-beta)+c/(x-gamma)

# we identify a,b,c by solving the following system:
Solutions=solve([LHS(0)-RHS(0,a,b,c),LHS(1)-RHS(1,a,b,c),LHS(2)-RHS(2,a,b,c)],[a,b,c]
print(Solutions)
astar=Solutions[a]
bstar=Solutions[b]
cstar=Solutions[c]
#anum=N(astar)
#bnum=N(bstar)
#cnum=N(cstar)

# to get a nice formula:
var('x')
print(latex(RHS(x,astar,bstar,cstar)))
```

```
{c: -1/168 + sqrt(7)/168, b: -sqrt(7)/168 - 1/168, a: 1/84}
\frac{- \frac{\sqrt{7}}{168} - \frac{1}{168}}{x - 4 + \sqrt{7}} + \frac
{- \frac{1}{168} + \frac{\sqrt{7}}{168}}{x - 4 - \sqrt{7}} + \frac{1}{8
4 \left(x + 3\right)}
```