Symbolic computation 3: Linear recurrences

Table of contents

- A periodic sequence
- A number theory theorem with SymPy
- A linear recurrence (almost) by hand
- Solving linear recurrences with SymPy
  - The function `rsolve`
  - Application 1
  - Application 2

Exercise 1. A periodic sequence
Do it yourself. Let \( a \notin \{1, -1\} \) be a real number, we define the sequence \((u_n)_{n \geq 0}\) as follows: \( u_0 = a \) and for \( n \geq 1 \)
\[
u_n = \frac{1 + u_{n-1}}{1 - u_{n-1}}.
\]

Using SymPy, prove that for every \( a \) the sequence \((u_n)\) is periodic.

```python
# We compute the first few terms:
def InductionFormula(x):
    return (1+x)/(1-x)

var('a')
Sequence=[a]
print('u_0 = a')
for i in range(1,12):
    Sequence.append(simplify(InductionFormula(Sequence[-1])))
    print('u_{%d} = %s' % (i, str(Sequence[-1])))
```

\[
\begin{align*}
  u_0 &= a \\
  u_1 &= -(a + 1)/(a - 1) \\
  u_2 &= -1/a \\
  u_3 &= (a - 1)/(a + 1) \\
  u_4 &= a \\
  u_5 &= -(a + 1)/(a - 1) \\
  u_6 &= -1/a \\
  u_7 &= (a - 1)/(a + 1) \\
  u_8 &= a \\
  u_9 &= -(a + 1)/(a - 1) \\
  u_{10} &= -1/a \\
  u_{11} &= (a - 1)/(a + 1)
\end{align*}
```

Answers. According to the above script we have
\[
u_4 = u_0
\]
for every \( a \). Since \( u_n \) is defined by a simple recurrence of order one this suffices to prove that
\[
u_{n+4} = u_n \tag{\star}
\]
for every \( n \), i.e. \((u_n)\) is periodic with period at most 4. Indeed we can prove eq. (\star) by a simple induction:

- For \( n = 0 \) we trust SymPy.
- Assume eq. (\star) is true for some \( n \). We write
  \[
u_{n+1+4} = u_{n+5} = f(u_{n+4})
  \]
  where \( f(x) = (1 + x)/(1 - x) \). By the induction hypothesis
  \[
f(u_{n+4}) = f(u_n) = u_{n+1} \]
  and the proof is done.

---

Exercise 2. A number theory theorem with Sympy
Do it yourself.

1. With Sympy simplify the following expression:

\[
\left( \frac{x^6 + 45x^4 - 81x^2 + 27}{3x(x^2 + 3)} \right)^3 + \left( \frac{-x^4 + 30x^2 - 9}{3x(x^2 + 3)} \right)^3 + \left( \frac{-12x^3 + 36x}{(x^2 + 3)^2} \right)^3
\]

2. Prove the following:

Theorem. Every integer \( n \) can be written as the sum of three cubes of rational numbers.

To save you time in the proof you can use the following notations:

\[
\begin{align*}
p &= x^6 + 45x^4 - 81x^2 + 27, \\
q &= 3x(x^2 + 3)^2, \\
r &= -x^4 + 30x^2 - 9, \\
s &= 3x(x^2 + 3), \\
t &= -12x^3 + 36x, \\
u &= (x^2 + 3)^2
\end{align*}
\]

```python
var('x')
A=(x**6+45*x**4-81*x**2+27)/(3*x*(x**2+3)**2)
B=(-x**4+30*x**2-9)/(3*x*(x**2+3))
C=(-12*x**3+36*x)/((x**2+3)**2)
simplify(A**3+B**3+C**3)
```

\( 8x \)

Answers. Let \( x \) be an integer. The above script shows that

\[
8x = (p/q)^3 + (r/s)^3 + (t/u)^3,
\]

where \( p, q, r, s, t, u \) are integers. Thus

\[
x = (p/2q)^3 + (r/2s)^3 + (t/2u)^3.
\]

Therefore \( x \) can be written as a sum of cubes of rationals.

**Exercise 3. Solving a recurrence (almost) by hand**
Do it yourself. Let \((u_n)\) be the sequence defined by
\[
\begin{align*}
  u_0 &= 1, \\
  u_n &= 2u_{n-1} + 3n^2 & (\forall n \geq 1)
\end{align*}
\]

The goal of the exercise is to find a formula for \(u_n\).

For this purpose we introduce the sequence \((v_n)\) defined by
\[
\begin{align*}
  v_0 &= u_0, v_1 = u_1, v_2 = u_2, v_3 = u_3 \\
  v_n &= \alpha 2^n + \alpha n^2 + bn + c & (\forall n \geq 0)
\end{align*}
\]
for some parameters \(\alpha, a, b, c\).

1. Use \texttt{SymPy} to find \(\alpha, a, b, c\).

To solve a system of equations with \texttt{SymPy} with unknowns \(x, y\), the syntax is
\[
\texttt{solve([[x-y-2,3*y+x],[x,y]])}
\]

2. Prove with \texttt{SymPy} that this sequence \((v_n)\) satisfies the recurrence \((\star)\): for every \(n\) we have
\[
v_n = 2v_{n-1} + 3n^2.
\]

\[
\begin{align*}
\texttt{# ------------- Question 1 --------------} \\
\texttt{var('n', integer=True)} \\
\texttt{var('alpha a b c')} \\
\texttt{def v(n, alpha, a, b, c):} \\
\texttt{\quad return alpha*2**n + a*n**2 + b*n + c} \\
\texttt{def u(n):} \\
\texttt{\quad if n==0:} \\
\texttt{\quad \quad return 1} \\
\texttt{\quad else:} \\
\texttt{\quad \quad return u(n-1)*2*3+n+n} \\
\texttt{Solutions=solve(} \\
\texttt{\quad # Quantities which are to be zero:} \\
\texttt{\quad [v(0, alpha, a, b, c)-u(0),} \\
\texttt{\quad \quad # initial condition} \\
\texttt{\quad v(1, alpha, a, b, c)-u(1),} \\
\texttt{\quad v(2, alpha, a, b, c)-u(2),} \\
\texttt{\quad v(3, alpha, a, b, c)-u(3),} \\
\texttt{\quad ]},} \\
\texttt{\quad # Unknowns:} \\
\texttt{\quad [alpha, a, b, c]} \\
\texttt{\quad )} \\
\texttt{print(Solutions)} \\
\texttt{\quad \{c: -18, alpha: 19, b: -12, a: -3\}}
\end{align*}
\]

\textbf{Answers.} According to the above script we get
\(\alpha = 19, \quad a = -3, \quad b = -12, \quad c = -18\).

With this choice of parameters, \((u_n)\) and \((v_n)\) coincide for the first values.
Answers. Question 2. According to the above script we proved that the formula is true: for every \( n \) we have that
\[
v_n - 2v_{n-1} = 3n^2.
\]
Finally we have proved that the solution of (\( \star \)) is given by
\[
v_n = 19 \times 2^n - 3n^2 - 12n - 18.
\]

**Solving recurrences with SymPy: rsolve**
The function \texttt{rsolve}

We will now see how to use Sympy to obtain explicit formulas for some sequences defined by linear recurrences. More precisely, we will see how to obtain an explicit formula for \( u_n \) in two cases:

1. **Linear recurrence of order one**: this is a sequence \((u_n)_{n \geq 0}\) is defined by
   
   \[
   \begin{aligned}
   u_0 &= a, \\
   u_n &= \alpha u_{n-1} + f(n), \quad (n \geq 1),
   \end{aligned}
   \]
   where \(a, \alpha\) are some given constants and \(f\) is an arbitrary function.

2. **Linear recurrence of order two**: this is a sequence \((u_n)_{n \geq 0}\) is defined by
   
   \[
   \begin{aligned}
   u_0 &= a, \\
   u_1 &= b, \\
   u_n &= \alpha u_{n-1} + \beta u_{n-2} + f(n), \quad (n \geq 2),
   \end{aligned}
   \]
   where \(a, b, \alpha, \beta\) are some given constants and \(f\) is an arbitrary function.

Some known examples fit in this settings:

1. Geometric sequences: \(u_0 = a, u_n = ru_{n-1}\).
2. Arithmetic sequences: \(u_0 = a, u_n = u_{n-1} + r\).
3. The Fibonacci sequence: \(F_1 = 1, F_2 = 1, F_n = F_{n-1} + F_{n-2}\).

The following script shows how to solve the linear recurrence of a geometric sequence, using function \texttt{rsolve}:
Exercise 4. Application: Fibonacci with \texttt{rsolve}

Do it yourself.

1. Use \texttt{SymPy} to solve the recurrence of the Fibonacci sequence and find an explicit formula for \(F_n\).

2. (Theory) Use the formula obtained at Question 1 to prove that

\[
\lim_{n \to +\infty} \frac{F_n}{F_{n-1}} = \varphi = \frac{1 + \sqrt{5}}{2}.
\]

\textit{(Hint: Set } \theta = (1 - \sqrt{5})/2 \text{.)}
Answers.

1. We put the above answer in \LaTeX:

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

2. The factor \( \frac{\sqrt{3}}{5} \) cancels out so we obtain:

\[
\frac{F_n}{F_{n-1}} = \frac{\varphi^n - \theta^n}{\varphi^{n-1} - \theta^{n-1}},
\]

\[
= \frac{\varphi - \theta(\varphi\theta)^{n-1}}{1 - \theta(\varphi\theta)^{n-1}},
\]

(we divide everything by \( \varphi^{n-1}. \))

Since \( |\varphi/\theta| < 1 \) the numerator tends to \( \varphi \), the denominator to one.

Remark. The output of \texttt{rsolve} is an expression which depends on the symbolic variable \( n \). If we want to evaluate this expression (for instance for \( n = 10 \)) we must write:

\[
\text{Value = ExplicitFormula.subs(n,10)}
\]

\[
\text{print('10th Fibonacci number = '+str(Value))}
\]

\[
\text{print('After simplification : '+str(simplify(Value)))}
\]

10th Fibonacci number = sqrt(5)*((-sqrt(5) + 1)**10 + (1 + sqrt(5))**10)/5120
After simplification : 55

Exercise 5. Application: Another example of order two

Do it yourself. Let \( (J_n) \) be defined by

\[
\begin{align*}
J_0 &= 1 \\
J_1 &= 2 \\
J_n &= 2J_{n-2} + 5 & \text{for every } n \geq 2
\end{align*}
\]

Use \texttt{rsolve} to find an explicit formula for \( J_n \).
The explicit formula for \( J_n \) is

\[
2^{\frac{n}{2} - 2} \left(7 \sqrt{2} + 12\right) + \frac{(-\sqrt{2})^n}{4} \left(-7 \sqrt{2} + 12\right) - 5
\]

**Do it yourself.** (Theory) According to the formula found by **SymPy**, what happens when \( n \to +\infty \) for the sequence \( \frac{J_n}{\sqrt{2}} \) ?

**Answers.** We rewrite the above formula as

\[
J_n = \sqrt{2}^{\frac{n}{2}} \frac{1}{4} \left(2 \sqrt{2} + 5 + (-1)^n(-2 \sqrt{2} + 5)\right) - 1.
\]

Therefore \( \frac{J_n}{\sqrt{2}} \) oscillates between \( 10/4 = 5/2 \) and \( 4 \sqrt{2}/4 = \sqrt{2} \).