A small library for the continued fraction expansion of rational numbers

Group N

Abstract

We present `continued_fractions`, a small Python library that computes the continued fraction expansion of a rational number and the corresponding list of convergents. The library is built on top of `sympy` [3] so that every iterate is held exactly. We illustrate the library on the rational approximations $3/2$, $22/7$ and $355/113$ of π, and we recover the Fibonacci-ratio approximations of the golden ratio. The documentation follows the Diataxis framework [4] and the conventions of the open-source textbook *Python for Mathematics* [2].

1 Introduction

The continued fraction expansion of a real number r is the (possibly infinite) sequence of integers $a_0, a_1, a_2, \ldots$ such that

$$r = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}}.$$

For a rational number the expansion terminates, while for an irrational number it is infinite. The convergents are the rational numbers obtained by truncating the expansion after each step; they are known to give the best rational approximations to r in the sense that no fraction with a smaller denominator is closer [1].

We wanted a small library that we could use to inspect these expansions without losing precision to floating-point rounding. The standard library does not provide one. We chose to build on top of `sympy` so that the iterates stay symbolic; this is the same approach taken by the textbook *Python for Mathematics* [2].

We present a three-function library, `continued_fractions`; we test it on the well-known rational approximations $22/7$ and $355/113$ of π; and we recover the ratios of consecutive Fibonacci numbers as the convergents of the golden ratio.

2 The library

The library lives in the single module `continued_fractions.py` and exposes three functions.

`continued_fraction(rational_number, maximum_terms)` computes the expansion of a rational by the Euclidean algorithm: at each step we take the integer part of what is left, subtract it, and invert the fractional part. The loop terminates as soon as the remainder is zero, so we never run past the natural end of the expansion.

`convergents(coefficients)` turns a list of integer coefficients into the corresponding rationals. We use the standard recursion

$$\frac{p_n}{q_n} = \frac{a_n p_{n-1} + p_{n-2}}{a_n q_{n-1} + q_{n-2}}.$$

This avoids rebuilding the nested fraction from scratch at each step.

`golden_ratio_convergents(number_of_terms)` returns the first convergents of the expansion $[1; 1, 1, 1, \ldots]$. We use this as a worked example because the convergents are exactly the ratios of consecutive Fibonacci numbers.

3 Worked examples

Two common rational approximations of π are 22/7 and 355/113. Their expansions are $[3; 7]$ and $[3; 7, 16]$ respectively, and the convergents recover the original rationals exactly, as shown in Table 1.

Number	Expansion	Convergents
22/7	$[3; 7]$	$3, 22/7$
355/113	$[3; 7, 16]$	$3, 22/7, 355/113$

Table 1: **Two rational approximations of π.** For each rational we show the continued fraction expansion and the corresponding list of convergents. The last convergent agrees with the input rational exactly.

The continued fraction $[1; 1, 1, 1, \ldots]$ gives the convergents $1, 2, 3/2, 5/3, 8/5, 13/8, \ldots$, which are ratios of consecutive Fibonacci numbers. The twentieth convergent agrees with $\varphi = (1 + \sqrt{5})/2$ to within 10^{-8}.

4 Discussion

The Euclidean step inverts a fraction, which on floating-point arithmetic introduces a small error at each iterate. The error compounds; by the third or fourth step it is large enough to affect the integer part. Working with `sympy.Rational` sidesteps this entirely. The cost is that the iterates can become large symbolic fractions, but for the inputs we care about (rationals with small denominators) this is not an issue.

We only handle rational inputs. A natural extension would be to handle the periodic continued fractions of quadratic irrationals such as $\sqrt{2} = [1; 2, 2, 2, \ldots]$, which have a known closed form. This is a direction for future work.

5 Conclusion

We have presented `continued_fractions`, a small Python library for the continued fraction expansion of rational numbers. The library recovers the standard rational approximations of π, and the convergents of the golden ratio match the ratios of consecutive Fibonacci numbers to within tolerance.

References

[1] Aleksandr Ya. Khinchin. *Continued Fractions.* University of Chicago Press, 1964.

[2] Vincent Knight. Python for Mathematics, 2024.

[3] Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, AmiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, et al. SymPy: symbolic computing in Python. *PeerJ Computer Science*, 3:e103, 2017.

[4] Daniele Procida. Diátaxis: a systematic framework for technical documentation authoring, 2017.