

Payment caps and negative amortization



MortgageMath
Precise Loan Amortization in Python

mortgagemath 0.5.2 · rendered 2026-05-02

The scenario

A **payment cap** is a different mechanic from a **rate cap**. A rate cap bounds the *interest rate*; a payment cap bounds the *dollar payment* directly, regardless of what the new rate would otherwise produce.

When a payment cap binds *and* the new periodic interest exceeds the capped payment, the unpaid interest is **capitalized** into the balance — the loan goes into **negative amortization**: the borrower is paying less than the interest accrual, the balance grows, and the per-row `Installment.principal` is **negative**.

The library models this with one optional field on `RateChange`: `payment_cap_factor` (e.g., `Decimal("1.075")` for a 7.5% annual cap). Validated against the ProEducate ARM Payment Caps worked example.

ProEducate loan terms

Term	Value
Principal	\$65,000
Term	30 years (360 monthly payments)
Year 1 rate	10%
Year 2 rate	12%
Annual payment cap	7.5% (factor 1.075)
Convention	30/360, ROUND_HALF_UP, round-each-balance

Encoded as LoanParams

```
from decimal import Decimal
from mortgagemath import (
    LoanParams, RateChange, PaymentRounding, BalanceTracking,
    amortization_schedule,
)

loan = LoanParams(
    principal=Decimal("65000"),
    annual_rate=Decimal("10"),
    term_months=360,
    payment_rounding=PaymentRounding.ROUND_HALF_UP,
    interest_rounding=PaymentRounding.ROUND_HALF_UP,
    balance_tracking=BalanceTracking.ROUND_EACH,
    rate_schedule=(
        RateChange(
            effective_payment_number=13,
            new_annual_rate=Decimal("12"),
```

```

        payment_cap_factor=Decimal("1.075"), # 7.5% annual cap
    ),
)
sched = amortization_schedule(loan)

```

How the cap binds

Quantity	Value
Year 1 P&I (closed-form at 10%)	\$570.42
Year 1 ending balance (after pmt 12)	\$64,638.72
Year 2 uncapped recast (closed-form at 12% on remaining 348)	\$667.30
Annual cap (year 1 P&I × 1.075)	\$613.20
Year 2 P&I used (= min(uncapped, cap))	\$613.20
Year 2 monthly interest (12% / 12 × balance)	\$646.39
Year 2 monthly principal (= payment – interest)	\$–33.19 (negative)

Around the rate-change boundary

Pmt	Rate	Payment	Interest	Principal	Balance
10	10%	\$570.42	\$539.44	\$30.98	\$64,701.46
11	10%	\$570.42	\$539.18	\$31.24	\$64,670.22
12	10%	\$570.42	\$538.92	\$31.50	\$64,638.72
13	12%	\$613.20	\$646.39	\$–33.19	\$64,671.91
14	12%	\$613.20	\$646.72	\$–33.52	\$64,705.43
15	12%	\$613.20	\$647.05	\$–33.85	\$64,739.28
24	12%	\$613.20	\$650.23	\$–37.03	\$65,059.62
25	12%	\$613.20	\$650.60	\$–37.40	\$65,097.02

The transition: month 12 (last payment at year-1 rate) ends with balance \$64,638.72. Month 13 applies the new 12% rate to that balance — interest is \$646.39, but the capped payment is only \$613.20. The \$33.19 gap is added to the balance, which now grows each month while the cap is binding.

Cumulative consequence

- Months in negative amortization (year 2): **12**
- Total interest capitalized into balance over year 2: **\$420.90**
- Balance at end of year 1: **\$64,638.72**
- Balance at end of year 2: **\$65,059.62**
- Net balance growth during year 2: **\$420.90**

Every value above reproduces the ProEducate published worked example to the cent — including the principal column being signed negative during negative amortization. The per-row invariant $\text{principal} + \text{interest} == \text{payment}$ continues to hold.

API and CLI

Python API:

```
RateChange(  
    effective_payment_number=13,  
    new_annual_rate=Decimal("12"),  
    payment_cap_factor=Decimal("1.075"),  
)
```

CLI:

```
mortgagemath schedule --principal 65000 --rate 10 --term-months 360 \  
    --payment-rounding ROUND_HALF_UP --interest-rounding ROUND_HALF_UP \  
    --rate-change 13:12:cap=1.075 --format csv
```

Source

ProEducate, *ARM Payment Caps* — consumer-handbook lineage, mirrored at mortgagesfinancingandcredit.org. Fixture: `tests/schedules/proeducate_arm_pmt_cap_65k_10pct_to_12pct_360mo.{toml,csv}`.

The textbook canonical example (Goldstein 12e §10.4 Example 14, a \$200,000 5/1 ARM with 8% payment cap at year 7) describes the same mechanic but is **not** committed as a fixture: Goldstein’s published balances are computed via closed-form PV math at the rounded payment, which drifts 13–26¢ from the library’s row-by-row schedule. ProEducate is the cents-level published-source fixture that validates the library’s payment-cap implementation.