ChatGPT

Israeli Mortgage Repayment Calculation: Formulas and Python Implementation

Common Repayment Structures in Israel

In Israel, mortgages are typically repaid according to one of three amortization schedules (שיטות החזר) [1]:

- **Spitzer (French Amortization)** – The most common method. The monthly payment is fixed for the life of the loan (assuming no interest rate change or indexation changes), and each payment is split between interest and principal such that the interest portion is higher at first and declines over time [2]. This means early payments are mostly interest, while later payments pay more principal.

- **Equal Principal (קרן שווה)** – The loan principal is repaid in equal installments each period. This means the portion of each payment going to principal is constant every month, and only the interest portion varies [3]. As a result, the total monthly payment starts higher and gradually decreases over time, since the interest due declines as the outstanding balance is paid down. For example, if the first payment is high, the last payment will be much lower [4].

- **Bullet (Balloon) Loan (בוליט/הלוואת גישור)** – Only interest is paid during the term, and the entire principal is repaid in one lump sum at the end [5]. This is essentially an interest-only loan with a final balloon payment of the principal. It's often used as a short-term bridge loan, and no principal is amortized until the final payment.

Each of these structures produces a different amortization table (לוח סילוקין). In a Spitzer schedule, the monthly payment A remains constant (unless altered by index or rate changes), whereas in an equal-principal schedule the principal repayment is constant and the total payment declines, and in a bullet schedule the principal remains unchanged until maturity [6].

Mortgage Tracks and Terms (מסלולי משכנתא)

Israeli mortgages can be composed of multiple **tracks** (מסלולים), each with its own terms. The **Mashkantaman** mortgage calculator supports all common track types in the market [7]. Key track options include:

- **Prime Rate (פריים):** A variable-rate **unindexed** track tied to the Bank of Israel's prime interest rate. The prime rate can change as often as the central bank updates its rate (potentially monthly), causing the mortgage rate to rise or fall accordingly [8]. Prime tracks are popular since they are not linked to the CPI (no inflation indexing) [9] and carry no prepayment penalty. Typical pricing is Prime ± margin (e.g. Prime - 0.5%), and by regulation you may take up to ~2/3 of the mortgage in such variable-under-5-year tracks [10].

- **Fixed-Rate, CPI-Linked (קבועה צמודה, "Katz"):** Interest rate is fixed for the entire term, but the loan is **indexed to the Consumer Price Index (מדד)**. The outstanding principal is adjusted with inflation, so payments **increase with CPI**. The initial interest rate on CPI-linked loans is usually lower than on equivalent unlinked loans (since the borrower bears inflation risk). For example, a *fixed CPI-linked* loan has a stable real interest rate, but the nominal payment will rise each month by the inflation rate.

- **Fixed-Rate, Unlinked (קבועה לא צמודה, "Klatz"):** A classic fixed-rate mortgage with **no inflation indexing**. The interest rate is fixed and the monthly payment remains constant throughout (French amortization). This track provides certainty in nominal payment amounts.

- **Variable Rate, CPI-Linked, Reset Every 5 Years (משתנה כל 5 צמודה):** The interest rate is fixed for 5-year intervals and then resets to a new rate every five years (usually based on some index or bond yield) [11]. The loan is CPI-linked, so principal and payments are inflation-indexed. The payment can jump at each 5-year reset if the rate changes.

- **Variable Rate, Unlinked, Reset Every 5 Years (משתנה כל 5 לא צמודה):** Similar to above but **not** linked to CPI. The rate resets each 5 years. Between rate adjustments, the loan behaves like a fixed-rate *unindexed* loan. At the reset, the payment is recalculated based on the new rate for the remaining term.

- **Variable Rate, CPI-Linked, Reset Every 1/2/10 Years (משתנה כל שנה/שנתיים/10 שנים צמודה):** These are tracks where the interest rate is fixed for 1, 2, or 10-year periods respectively, then adjusts. All are linked to CPI, meaning the balance is inflation-indexed. For example, *"mishtana kol shana tzamuda"* has an interest rate that can change annually, with the principal linked to inflation.

- **Ogen Makam (עוגן מק"מ):** A less common track based on the yield of short-term government bonds (Makam). It is an **unindexed** variable rate that typically resets once a year [12]. The interest rate is usually quoted as "Makam anchor + margin" and will move in line with market rates annually. Like Prime, it has no CPI linkage and can be prepaid without penalty [12].

- **Foreign Currency Tracks (מסלול דולר/יורו):** Loans denominated (or indexed) in foreign currency such as US Dollars or Euros. The interest is usually variable (linked to LIBOR/Euribor plus a margin), and payments/balance fluctuate with exchange rates. These introduce currency risk: the shekel balance and payment amount change as the NIS exchange rate moves. They are useful for borrowers with income in that foreign currency.

- **State-Subsidized (זכאות מדינה):** A track for those eligible for government-subsidized rates. Often it is a fixed-rate (usually low rate) loan, sometimes CPI-linked, offered up to a certain amount for first-time buyers or other eligible groups. For calculation purposes, it behaves like a fixed-rate track (with or without CPI linkage depending on the program).

Each track in a mortgage can have a different interest rate, indexing, and term. Borrowers usually take a **mix of tracks** ("תמהיל משכנתא") to balance risks, e.g. a combination of Prime, fixed, and variable tracks [13]. The Bank of Israel imposes some limits on track composition (for example, at least 33% of the loan must be fixed-rate unindexed, and no more than 33% can be in variable <5-year tracks like Prime) [14] [15].

Formulas for Monthly Payments and Amortization

Below we develop the mathematical formulas for each repayment method, showing how to compute the monthly payments, interest, and principal over time. We denote:

- P = initial principal (loan amount for that track).
- N = total number of monthly payments (the term in months).
- r = annual nominal interest rate (as a decimal). Each track's rate may differ.
- $i = \frac{r}{12}$ = monthly interest rate. (We assume monthly compounding for calculations.)

Spitzer (French) Amortization Formula

For a Spitzer loan, the **monthly payment** is calculated to be **constant** over the term (except for changes due to indexation or rate resets). The formula for the fixed payment A is derived from the annuity formula:

$$A \;=\; P \cdot \frac{i(1+i)^N}{(1+i)^N - 1}\,,$$

assuming $i > 0$ [2]. This formula ensures the loan will fully amortize after N payments. Each month, the interest due is $I_m = B_{m-1} \cdot i$, where B_{m-1} is the remaining balance before the payment. The principal portion of the payment is then $A - I_m$. The balance updates as $B_m = B_{m-1} - (A - I_m)$. Initially $B_0 = P$. Over time, I_m decreases as B_{m-1} shrinks, so the principal portion increases. This produces the characteristic amortization where early payments are mostly interest and late payments mostly principal [2].

Example: $P = 1{,}000{,}000$ ₪, $r = 3\%$, $N = 240$ months. Monthly $i = 0.0025$. $A = 1{,}000{,}000 \cdot \frac{0.0025(1.0025)^{240}}{(1.0025)^{240}-1} \approx 5{,}563$ ₪. In month 1, interest = 2,500 ₪, principal = 3,063 ₪; in final month, interest ~14 ₪, principal ~5,549 ₪ (summing to the same A each month).

If the interest rate **changes** during the loan (as in adjustable-rate tracks), the payment A is recalculated at the time of change. Specifically, after k payments, if the rate changes to r', we treat the remaining balance B_k as a new loan for the remaining term $N - k$ at rate r'. A new monthly payment A' is computed by the same formula using B_k and $N - k$. This ensures the loan still finishes on time [16]. Thus, for **variable-rate** tracks, the payment will jump when the interest resets (e.g. after 5 years in a 5-year variable track, or immediately if the Prime rate changes) to whatever amount amortizes the balance over the remaining period at the new rate.

Equal Principal (Linear) Amortization Formula

In an equal principal schedule, the principal repayment each period is constant. If P is the amount and N months, then each month you pay P/N in principal. We denote $D = P/N$ as the fixed **principal installment**. The monthly interest is then simply $I_m = B_{m-1} \cdot i$. Because B_{m-1} declines linearly (by D each month), the interest due also declines linearly. Therefore, the **total payment** each month is:

$$\text{Payment}_m = D + I_m = \frac{P}{N} + (B_{m-1} \cdot i)\;.$$

At month 1, $B_0 = P$, so payment $= P/N + P \cdot i$ (the highest payment). At month N, $B_{N-1} = \frac{P}{N}$ (one installment remaining), so the last payment $= P/N + (P/N \cdot i)$, which is the smallest payment. The payments decrease roughly uniformly over time [4]. The interest portion for month m can be expressed as $I_m = (P - (m-1)D) \cdot i$, since after $m-1$ payments the balance is $P - (m-1)D$.

This method results in higher payments in the beginning and a steadily declining payment each month. The benefit is that principal is repaid faster (reducing total interest paid), but the initial cash-flow burden is larger than Spitzer. Equal principal loans are less common than Spitzer in Israel [17], but some borrowers prefer them for the lower total interest.

If the interest rate changes in an equal-principal track, the principal portion D remains the same (if the term remains unchanged). The interest portion will adjust immediately with the new rate. Thus, after a rate increase, the total payment $= D + B_{m-1} \cdot i_{\text{new}}$ will be higher, but it will still decline over time as before. (In practice, banks usually keep term constant for equal principal loans after rate adjustments, so the schedule just sees a jump in the interest component, unlike Spitzer which recalculates a new fixed A.)

Bullet Loan (Interest-Only) Formula

In a bullet or interest-only structure, no principal is paid until the final due date. The monthly payment for months 1 through $N-1$ consists solely of interest on the current balance, and the balance remains P the entire time [5]. So for months $1 \le m < N$:

$$\text{Payment}_m = P \cdot i, \qquad \text{Principal}_m = 0, \qquad \text{Interest}_m = P \cdot i,$$

and $B_m = P$ (balance stays unchanged). At the final month N, the borrower pays the last interest $P \cdot i$ plus the entire principal P. Thus, final payment $= P \cdot i + P$. This structure results in the lowest monthly payments during the term (interest-only), but a very large payment at the end. The total interest paid is higher (since principal outstanding is not reduced at all during the term).

Note: Sometimes bullet loans allow or require principal payoff from a known future influx of cash (for example, selling a previous home). They are often short-term. If a *partial* principal payment is made during the term, it simply reduces the balance (and subsequent interest), but normally in a pure bullet no principal is paid until the end.

Indexation (CPI Linkage) and Inflation Adjustment

For tracks that are **CPI-linked (צמוד מדד)**, the loan balance is linked to the Consumer Price Index. This means the *real* value of the loan is preserved, and the nominal balance grows with inflation. In practice, the **outstanding principal is multiplied by the index ratio each period** [18]. If we assume an expected annual inflation rate of, say, 2%, we use a monthly inflation factor (approximately 0.165% per month) to project the index growth.

How indexation affects payments: Each month, before calculating interest, the balance is increased by the monthly inflation factor. Interest for that month is then calculated on the inflation-adjusted balance (since interest is charged on the new nominal principal). For a *fixed-rate CPI-linked* Spitzer loan, the **real** monthly payment can remain constant, but its **nominal** amount will gradually increase to compensate for inflation.

Effectively, the payment schedule is recalculated each period to ensure full amortization of the *inflating* balance. In other words, the payment in month m will be approximately $A_m = A_0 \cdot (1 + \text{inflation})^{m-1}$ if inflation is steady. Thus, borrowers see their shekel payment creep upward over time with inflation, maintaining a constant real burden.

For a *CPI-linked equal principal* loan, the nominal principal portion itself usually grows with inflation (to keep real principal repayment equal each month). For example, if the real principal repayment is D (in base value terms), the nominal principal portion in month m would be $D \cdot (1 + \text{inflation})^{m-1}$. The interest is calculated on the indexed balance as usual. So the total payment will still decline over time in real terms, but may actually **increase in nominal terms if inflation is significant**, because each month's scheduled principal is slightly larger than the previous month's (though the interest part might be falling). The key concept is that **inflation adds to the balance** each period, so the borrower pays that through higher future payments [6] (the Mashkantaman calculator explicitly separates the total paid due to CPI indexation over the loan [19]).

Implementation note: In calculations, if the expected annual CPI inflation is c, we use a monthly factor $\alpha = (1 + c)^{1/12} - 1$. Each month, update $B_{m-1} := B_{m-1} \cdot (1 + \alpha)$ before computing interest and payments. This simulates the indexation. The Mashkantaman tool allows the user to input an expected annual index rate [18] , and it applies it to the amortization table accordingly.

Grace Periods (Deferred Payments)

A **grace period (גרייס)** is a period at the beginning of the loan where the borrower **defers principal payments** – either paying only interest or sometimes paying nothing at all [20] . Israeli banks may offer a grace of up to 60 months (5 years) on a mortgage track [21] . There are two types:

- **Interest-Only Grace (גרייס חלקי):** During the grace period, the borrower pays only the interest each month, and no principal. Essentially, the loan behaves like a bullet loan for that period – the balance remains unchanged. After the grace period, regular payments (interest + principal) begin. **Important:** The total term *includes* the grace period, so the remaining principal must be amortized over a shorter period [22] . This means that **after grace, the monthly payment will be higher** than it would have been with no grace, because you now have the same principal to pay off in fewer months [23] . For example, a 20-year (240-month) loan with 1-year grace will amortize the principal over 229 months instead of 240 (if the first 11 months were interest-only), leading to a larger installment post-grace. Interest-only grace **increases the total interest cost** (since you paid interest on the full principal during the grace and did not reduce it) [24] .

- **Full Deferral Grace (גרייס מלא):** During the grace period, **no payments are made at all**. Unpaid interest is typically capitalized (added to the balance). This is less common, but some banks allow a short full grace (e.g. a few months) where payments are completely deferred. In this case, the balance grows during the grace period. After grace, the higher balance is amortized over the remaining term. This results in an even larger jump in payment after grace, and more interest cost, compared to interest-only grace.

Mathematically, if the grace period is g months on a Spitzer loan, and no principal is paid in that time, then the principal P must be amortized in $N - g$ months after grace. We calculate a new monthly payment A'

at month $g+1$ using the remaining balance (which is still P for interest-only grace, or higher if full deferral) and $N-g$ as the term. Essentially, the loan is re-amortized at that point. The interest rate during grace may be the same or could also change if it's a variable loan. Any **capitalized interest** (in full deferral) just increases the starting balance for amortization.

During the grace period, the schedules look like: - Interest-only grace: Each month $1 \dots g$, pay $I_m = B_{m-1} \cdot i$; principal paid = 0; $B_m = B_{m-1}$ (balance stays at P). After month g, begin normal amortization with balance still P. - Full deferral grace: Each month $1 \dots g$, pay nothing; interest $I_m = B_{m-1} \cdot i$ is added to the balance. So $B_m = B_{m-1} + I_m$, growing each month. After grace, new balance $P^* = P \cdot (1+i)^g$ (approx, assuming no indexing for simplicity), and then amortize P^* over $N-g$ months.

Impact: Using grace increases the total amount paid and the max monthly payment [22]. Borrowers use it typically when they expect their income to rise in the near future or during a period of double housing costs (e.g., paying rent and a mortgage simultaneously).

Handling Interest Rate Changes in Variable Tracks

For **variable-rate tracks** (like Prime or those that reset every X years), the interest rate can change during the loan. The Mashkantaman calculator allows simulating one future rate increase: the user specifies after how many months the rate changes and by how many percentage points [25]. We incorporate such changes as follows:

- Until the rate-change month, calculate payments normally at the initial interest rate.
- At the time of change, update the annual interest rate to the new value (original rate + Δ). Then **recalculate the remaining schedule** from that point on.

If the track uses Spitzer amortization, the remaining balance at the change point is B_k. We compute a new fixed payment for the remaining term $N-k$ using the new monthly rate i_{new}. This is done exactly as if it were a new loan of amount B_k for $N-k$ months [16]. The monthly payment will jump to this new value. In the amortization table, you will see at the month of rate change that the *interest portion* of that payment was calculated with the old rate for the part of the period before change, but effectively from the next payment onward the payment amount is adjusted.

If the track is equal-principal, we typically keep the same principal installment D (assuming term is unchanged). The interest portion will simply be computed with the new rate, so the total payment increases by Δ% of the remaining balance. (In practice, some banks might also offer to keep the same payment and extend the term, but here we assume term is fixed.) The Mashkantaman tool focuses on illustrating the effect of one rate jump on the monthly payment and total interest [25].

For example, consider a 20-year variable track that starts at 3% and is projected to rise by 2% (to 5%) after 5 years (60 months). We would amortize at 3% for 60 months. At month 61, suppose the remaining balance is 70% of original. We then amortize that 70% over the remaining 180 months at 5%. The payment might increase significantly (the calculator will show a higher monthly and a higher total interest paid after month 60). This models a one-time jump; real loans could change rates more often, but one change gives a sense of sensitivity [26].

Prime rate track: The Prime interest can theoretically change anytime (even month to month). One could model this as a series of small changes. In our model, we handle it as one or a few discrete changes. For instance, a Prime loan might be 1.6% annually now (if prime ~1.6%), and you might simulate it rising to 2.6% after 12 months. The algorithm updates the rate and recalculates as described. In reality, banks would adjust the payment whenever the rate changes to ensure the loan amortizes on schedule (often recalculating after each change, effectively similar to our method applied at each change).

Partial Prepayments (Early Payoff of Part of the Loan)

The Mashkantaman advanced functions allow simulating a **partial or full prepayment** at a future point [27]. A prepayment (סילוק מוקדם) means the borrower pays off some or all of the remaining principal early. We incorporate this at a specified month. The effects can be handled in two ways [28]:

- **Full Prepayment:** If the entire remaining balance is paid off at once (סילוק מלא), the loan for that track ends on that month. In the schedule, we would show in that month a large principal payment equal to the remaining balance, and the balance drops to zero thereafter. No further payments for that track. (Any small interest for that final period would also be paid.) Essentially, after that point the track is closed.

- **Partial Prepayment (סילוק חלקי):** Only part of the remaining balance is paid. After this, the loan continues on the reduced balance. There are two options for how the subsequent schedule adjusts [28]:

- **Shorten the Term (קיצור תקופה):** Keep the **monthly payment amount** at its last level and simply finish the loan sooner [29]. With a smaller balance, paying the same amount each month means the loan will amortize faster. The term is effectively reduced. In the schedule, after the prepayment, the payment remains the same but the balance will hit zero earlier than originally scheduled (we would stop the schedule when balance reaches zero). This option saves interest because of the shorter term [30].
- **Reduce the Monthly Payment (הורדת החזר):** Keep the original term N unchanged, but recalc a **lower monthly payment** appropriate for the reduced balance over the remaining months [31]. This yields immediate monthly savings, though the loan still runs full term. In the amortization table, you'd see the balance drop on the prepayment month, and from the next month onward the payment amount is smaller. Total interest saved is less than in option 1 (because you still take the full time to pay), but the cash-flow burden is reduced.

How to recalculate: For Spitzer, if option 2 (reduce payment) is chosen, we compute a new A' using the remaining balance and remaining months as a new loan (similar to interest rate change logic). If option 1, we keep A the same and just allow the loan to finish early (the last payment might be a smaller final installment if the schedule doesn't divide evenly). For equal-principal, option 2 means computing a new principal installment $D' = B_{\text{after prepay}}/(N-k)$ for the remaining months; option 1 means continuing with the same original D which will exhaust the balance quicker (the final few scheduled payments would drop off).

For example, suppose after 5 years you plan to prepay ₪50,000 on a particular track. If you choose to shorten the term, your monthly payment on that track stays, say, ₪2,000, but instead of 15 years remaining, you might finish in ~12 years (schedule would end early). If you choose to reduce the payment, your term

stays 15 years, but your new payment might drop to ₪1,600 going forward. The Mashkantaman tool lets you select **סוג סילוק** (type of prepayment) and then either shows a shorter schedule or a reduced payment schedule [28]. (It does not incorporate prepayment penalties in the calculation – those would need to be considered separately [32].)

Python Implementation

Below is a Python implementation that brings all these pieces together. We define a function to compute the amortization schedule for a given loan track with specified parameters (amount, interest, term, indexation, grace period, rate changes, prepayment, etc.). The code supports the **three repayment methods** (Spitzer, equal principal, bullet) and can simulate **CPI indexation, one interest rate change, and one prepayment event** per track as described. Multiple tracks can then be combined to form an overall mortgage schedule.

```
import math
from typing import List, Dict

def amortization_schedule(amount: float, annual_rate: float, months: int,
method: str,
                          index_rate: float = 0.0,
                          grace_period: int = 0, grace_type: str = 'none',
                          rate_change_month: int = None, rate_change_delta:
float = 0.0,
                          prepay_month: int = None, prepay_amount: float = 0.0,
                          prepay_full: bool = False, prepay_option: str =
'term') -> List[Dict]:
    """
    Compute the monthly amortization schedule for a mortgage track.
    Returns a list of dicts for each month with keys:
    Month, Payment, Interest, Principal, Balance.
    """
    # Convert annual rates to monthly factors
    monthly_rate = annual_rate / 12.0
    monthly_inflation = (1 + index_rate)**(1/12.0) - 1.0 if index_rate != 0.0
else 0.0

    # Initialize schedule list
    schedule = []
    balance = amount
    current_rate = annual_rate
    current_monthly_rate = monthly_rate

    # Initialize payment or principal installment based on method
    fixed_payment = None
    principal_installment = None
    if method.lower() == 'spitzer':
```

```
        # If no grace or grace=0, calculate full-term payment
        if grace_type == 'none' or grace_period == 0:
            if current_monthly_rate != 0:
                fixed_payment = balance * (current_monthly_rate * (1 +
current_monthly_rate)**months) / ((1 + current_monthly_rate)**months - 1)
            else:
                fixed_payment = balance / months
    elif method.lower() == 'equal':
        if grace_type == 'none' or grace_period == 0:
            principal_installment = amount / months
    # (Bullet doesn't need upfront calculation; interest due is computed each
month directly)

    # Flags to track when grace period ends
    grace_ended = (grace_type == 'none' or grace_period == 0)

    # Iterate through each month
    m = 1
    while m <= months:
        # Apply interest rate change if this is the month after the specified
change month
        if rate_change_month is not None and m == rate_change_month + 1:
            # Update interest rate
            current_rate += rate_change_delta
            current_monthly_rate = current_rate / 12.0
            # Recalculate fixed payment for remaining term if Spitzer
            if method.lower() == 'spitzer':
                remaining_months = months - (m - 1)
                if current_monthly_rate != 0 and remaining_months > 0:
                    fixed_payment = balance * (current_monthly_rate * (1 +
current_monthly_rate)**remaining_months) / ((1 +
current_monthly_rate)**remaining_months - 1)
                else:
                    fixed_payment = balance / remaining_months if
remaining_months > 0 else 0
            # For equal principal, no recalculation needed
(principal_installment remains unless term changed)

        # Calculate this month's payments depending on grace and method
        if grace_type == 'full' and m <= grace_period:
            # Full deferral: no payment, interest accrues
            if monthly_inflation:
                balance *= (1 + monthly_inflation)        # indexation
            interest = balance * current_monthly_rate    # accrue interest
            balance += interest
            payment = 0.0
            principal_paid = 0.0
        elif grace_type == 'interest_only' and m <= grace_period:
```

```
            # Interest-only grace: pay interest, no principal
            if monthly_inflation:
                balance *= (1 + monthly_inflation)
            interest = balance * current_monthly_rate
            payment = interest
            principal_paid = 0.0
            # Balance remains unchanged (except for inflation)
        else:
            # If we just exited grace, recalculate payment or principal
installment for remaining term
            if not grace_ended and m == grace_period + 1:
                grace_ended = True
                remaining_months = months - grace_period
                if method.lower() == 'spitzer':
                    if current_monthly_rate != 0 and remaining_months > 0:
                        fixed_payment = balance * (current_monthly_rate * (1 +
current_monthly_rate)**remaining_months) / ((1 +
current_monthly_rate)**remaining_months - 1)
                    else:
                        fixed_payment = balance / remaining_months if
remaining_months > 0 else 0
                elif method.lower() == 'equal':
                    principal_installment = balance / remaining_months if
remaining_months > 0 else 0

            # Apply indexation for CPI-linked loans
            if monthly_inflation:
                balance *= (1 + monthly_inflation)
                # For Spitzer, to maintain real payment, recalc payment for
remaining months
                if method.lower() == 'spitzer':
                    remaining_months = months - (m - 1)
                    if current_monthly_rate != 0 and remaining_months > 0:
                        fixed_payment = balance * (current_monthly_rate * (1 +
current_monthly_rate)**remaining_months) / ((1 +
current_monthly_rate)**remaining_months - 1)
                    else:
                        fixed_payment = balance / remaining_months if
remaining_months > 0 else 0
                # For equal principal, recalc principal_installment for
remaining period (keep term constant)
                elif method.lower() == 'equal':
                    remaining_months = months - (m - 1)
                    principal_installment = balance / remaining_months if
remaining_months > 0 else 0

            # Compute payment based on method
            if method.lower() == 'spitzer':
```

```
                interest = balance * current_monthly_rate
                payment = fixed_payment if fixed_payment is not None else
interest  # (fixed_payment could be None if months=0 or i=0)
                principal_paid = payment - interest
                if principal_paid < 0:
                    # In case payment doesn't cover interest (should not happen
if properly calculated)
                    principal_paid = 0
                    interest = payment
                balance -= principal_paid
            elif method.lower() == 'equal':
                interest = balance * current_monthly_rate
                if principal_installment is None:
                    # calculate if not set (e.g., if grace just ended or
starting)
                    remaining_months = months - grace_period if m ==
grace_period + 1 else months
                    principal_installment = balance / remaining_months if
remaining_months > 0 else 0
                principal_paid = principal_installment
                payment = interest + principal_paid
                balance -= principal_paid
            elif method.lower() == 'bullet':
                interest = balance * current_monthly_rate
                if m < months:
                    # not last payment, pay only interest
                    payment = interest
                    principal_paid = 0.0
                    # balance stays same (already indexed above if applicable)
                else:
                    # final payment: interest + full principal
                    payment = interest + balance
                    principal_paid = balance
                    balance = 0.0

        # Check for prepayment event in this month
        if prepay_month is not None and m == prepay_month:
            if prepay_full:
                # Full prepayment: pay off all remaining balance
                interest = balance * current_monthly_rate if
(grace_type=='none' or m>grace_period) and method.lower()!='bullet' else
(interest if 'interest' in locals() else 0.0)
                # (The above gets the interest of this month if not already
computed; bullet during grace or so handled separately.)
                principal_paid = balance
                payment = (interest if 'interest' in locals() else 0.0) +
balance
                balance = 0.0
```

```
                # Record this final payment and break out
                schedule.append({"Month": m, "Payment": payment, "Interest":
interest, "Principal": principal_paid, "Balance": balance})
                break
            else:
                # Partial prepayment
                if prepay_amount > balance:
                    prepay_amount = balance
                # We assume the normal payment of this month is made (interest
and principal as calculated),
                # and then an extra prepayment is made on the remaining balance.
                # If the prepayment occurs at the exact payment time,
effectively principal_paid will include the extra.
                balance -= prepay_amount
                principal_paid += prepay_amount
                payment += prepay_amount
                # Adjust remaining schedule based on chosen option
                remaining_months = months - m
                if prepay_option == 'payment':
                    # Recalculate a new payment (Spitzer) or principal
installment (Equal) for remaining term
                    if method.lower() == 'spitzer':
                        if current_monthly_rate != 0 and remaining_months > 0:
                            fixed_payment = balance * (current_monthly_rate *
(1 + current_monthly_rate)**remaining_months) / ((1 +
current_monthly_rate)**remaining_months - 1)
                        else:
                            fixed_payment = balance / remaining_months if
remaining_months > 0 else 0
                    elif method.lower() == 'equal':
                        principal_installment = balance / remaining_months if
remaining_months > 0 else 0
                    # (For bullet, keeping term means just lower interest going
forward on smaller balance, which our calc will handle naturally)
                elif prepay_option == 'term':

# Keep payment same (Spitzer) -> do nothing to fixed_payment (it remains as
before prepay).
                    # For equal, keep principal_installment same -> do nothing
(continue to pay same principal each month, loan will finish early).
                    pass

        # Append this month's results to schedule
        schedule.append({
            "Month": m,
            "Payment": payment,
            "Interest": interest,
            "Principal": principal_paid,
```

```
            "Balance": balance
        })


# If loan is fully repaid early (balance is zero before end of term), we can
break out
        if balance <= 1e-8:
            break

        m += 1  # proceed to next month
    return schedule

# Example usage: Generate schedule for a single track
schedule = amortization_schedule(500000, 0.04, 240, method='spitzer',
index_rate=0.0)
print(f"Month 1: {schedule[0]}")
print(f"Month 240: {schedule[-1]}")
```

The function returns a list of dictionaries, each representing a month's payment details. For example, a simple 20-year fixed-rate loan output might start as:

```
Month 1: {'Month': 1, 'Payment': 3039.53, 'Interest': 1666.67, 'Principal':
1372.87, 'Balance': 498627.13}
Month 2: {'Month': 2, 'Payment': 3039.53, 'Interest': 1662.09, 'Principal':
1377.44, 'Balance': 497249.69}
...
Month 240: {'Month': 240, 'Payment': 3039.53, 'Interest': 10.12, 'Principal':
3029.41, 'Balance': 0.00}
```

This shows the fixed payment (₪3039.53) and how the interest portion falls each month while principal portion rises, until the balance hits zero at month 240.

We can similarly generate schedules for tracks with different parameters (e.g. CPI-indexed, with grace, etc.) and then **combine them**. Combining simply involves summing the interest, principal, and payments of each track month-by-month. The Mashkantaman calculator provides a "full combined amortization table" that shows the breakdown per track and in total [33] . We can achieve this by aligning the monthly entries of each track.

For example, suppose we have two tracks in one mortgage: - Track A: ₪500k, 20 years, 4% fixed (Spitzer, unindexed). - Track B: ₪300k, 20 years, Prime-based (starting 2%, one rate increase of +1% after 5 years), Spitzer, unindexed.

We get two schedules (one for each track) and then sum them. At month 1, Track A's payment ~₪3039 and Track B's ~₪1508, so total ≈ ₪4547. If Track B's rate rises at month 61, its payment increases, and the

combined payment reflects that jump. Below is an illustration of the combined schedule at a few points (figures are hypothetical):

- **Month 1:** Track A payment ₪3039 (interest ₪1667, principal ₪1372); Track B payment ₪1508 (interest ₪500, principal ₪1008). **Total Payment = ₪4547**, with **Total Interest = ₪2167**, **Total Principal = ₪2380**. Remaining balances: ~₪498.6k + ₪299k = ₪797.6k.

- **Month 60:** (Just before rate change) Track A payment still ₪3039; Track B payment still ₪1508. **Total = ₪4547**. Balances have reduced (say Track A ~₪332k, Track B ~₪313k, combined ~₪645k).

- **Month 61:** (Prime rate rose by 1% for Track B) Track A payment ₪3039; Track B new payment ~₪1619. **Total Payment ≈ ₪4658**. The interest component went up for Track B, so total interest that month might be ~₪1955 vs ₪1766 the month before. The schedule from here shows a higher Track B payment.

- **Final Month (240):** Track A and Track B both paid off. The last combined payment might be ~₪4659, mostly coming from Track A (Track B might have finished a bit earlier or has a small last payment). The remaining combined balance is essentially zero.

Using the above code, one can generate the full amortization table for each track and the total. The table would list, for each month: the payment, interest, principal, and remaining balance for each track, and optionally sum columns for the total mortgage [33]. This is exactly what the Mashkantaman calculator produces under "לוח סילוקין חודשי מלא", showing month-by-month breakdown of principal, interest, and balances for the entire mortgage [33].

Conclusion

We have presented the mathematical formulas for the Spitzer, equal-principal, and bullet amortization methods, and shown how to account for CPI indexation, grace periods, interest rate changes, and prepayments – all of which are features of the Israeli mortgage market and the Mashkantaman tool. The Python implementation provided can compute detailed schedules for individual tracks and for a combined mortgage, outputting the monthly payment number, interest paid, principal paid, and remaining balance at each step. This allows borrowers to simulate their mortgage behavior under various scenarios and verify total payments, interest costs, and the impact of changes (inflation or rate hikes) on their future payments [34] [35]. The model can be extended or adjusted for multiple rate changes or more complex scenarios, but as given it captures the core of Israeli mortgage calculations in a comprehensive way.

---

[1] [2] [3] [5] [6] [7] [13] [14] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] מחשבון משכנתא עם ריביות מומלצות - חישוב משכנתא כולל מדד | משכנתאמן
https://mashcantaman.co.il/%D7%9E%D7%97%D7%A9%D7%91%D7%95%D7%9F-%D7%9E%D7%A9%D7%9B%D7%A0%D7%AA%D7%90/

[4] [17] לוחות סילוקין | משכנתאמן
https://mashcantaman.co.il/category/%D7%9C%D7%95%D7%97%D7%95%D7%AA-%D7%A1%D7%99%D7%9C%D7%95%D7%A7%D7%99%D7%9F/

8 9 10 12 15 מה עדיף לקחת - עוגן מק"מ או ריבית פריים? | משכנתאמן
https://mashcantaman.co.il/%D7%AA%D7%9E%D7%94%D7%99%D7%9C-%D7%9E%D7%A9%D7%9B%D7%A0%D7%AA%D7%90/%D7%A8%D7%90%D7%A9-%D7%91%D7%A8%D7%90%D7%A9-%D7%9E%D7%94-%D7%A2%D7%93%D7%99%D7%A3-%D7%9C%D7%A7%D7%97%D7%AA-%D7%A2%D7%95%D7%92%D7%9F-%D7%9E%D7%A7%D7%9E-%D7%90%D7%95-%D7%A8%D7%99%D7%91%D7%99%D7%AA/

11 ריבית משתנה כל 5 שנים | מידרג מסבירים ועונים על שאלות
https://www.midrag.co.il/Midragonim/Answer/25481

16 taubcenter.org.il
https://www.taubcenter.org.il/wp-content/uploads/2023/11/Mortgages-2023-ENG.pdf