

Technische Dokumentation

ALM-Simulator für Rentnerkassen

Beilage zur Diplomarbeit mit Finanzmathematischen Grundlagen und
Funktionsreferenz

`main_alm_simulation.py, sensitivity_analysis.py, rentnerbestand_generator.py, functions.R`

Inhalt

1.	Einleitung und Überblick.....	3
2.	Anlageklassen und Parameter	4
3.	Simulation korrelierter Renditen	5
4.	Bond-Renditemodell	6
5.	Sterblichkeitsmodellierung und Bestandsführung.....	8
6.	Barwertberechnung der Verpflichtungen $W(t)$	10
7.	Leibrentenbarwertfaktor — Detailberechnung	12
8.	Duration-Berechnung (Macaulay-Duration der Liabilities)	13
9.	Duration-Management und Cash Flow Matching	14
10.	Vermögensentwicklung und Deckungsgrad.....	15
11.	Spezialfunktionen	17
12.	Risikometriken und Optimierung.....	18
13.	Systematische Sensitivitätsanalyse (sensitivity_analysis.py)	19
14.	R-Analysefunktionen (functions.R / functions_parallel.R) — Kurzübersicht	20
15.	Referenz: Input-Variablen.....	23
16.	Referenz: Ergebnis-Spalten (mc_all_paths*.csv)	25

1. Einleitung und Überblick

Diese Dokumentation beschreibt die finanzmathematischen Grundlagen und Rechenschritte des ALM-Simulators (Asset-Liability Management) für Rentnerkassen. Der Simulator verwendet Monte-Carlo-Methoden, um die zukünftige Entwicklung von Vermögen (Assets) und Verpflichtungen (Liabilities) unter stochastischen Kapitalmarktszenarien zu modellieren.

Das zentrale Skript `main_alm_simulation.py` implementiert folgende Kernkomponenten:

- Korrelierte Rendite-Simulation mit Fat-Tail-Unterstützung (Gaussian Copula / Student-t)
- Realistisches Bond-Modell mit Coupon, Duration-Effekt, Pull-to-Par und Default-Risiko
- Aktuarielle Barwertberechnung der Verpflichtungen inkl. Ehegattenanwartschaft
- Macaulay-Duration der Liabilities mit Duration-Matching-Strategien
- Cash Flow Matching (CFM) mit Tranchen-basiertem Bond-Portfolio
- Sterblichkeitsmodellierung basierend auf Sterbetafeln
- Risikomanagement: Interest Rate Cap, Sonder-Rente, Sammelstiftung-Modus (Modellierung einer Gemeinschaftseinrichtung, vgl. Kap. 11.3)

2. Anlageklassen und Parameter

Das Modell umfasst 6 Anlageklassen. Der Index 0 (Zinsen) dient als Basiszinsfaktor und wird nicht direkt als Anlageklasse investiert, sondern steuert die Zinsstrukturkurve.

Index	Klasse	Deutsch	Parameter	Beschreibung
0	InterestRate	Zinsen	$\mu_0, \sigma_0, \text{slope}$	Basiszins (Cash-Rate), kein direktes Investment, yield slope
1	GovBonds	Staatsanleihen	D_Gov	Bond-Modell mit Duration
2	CorpBonds	Unternehmensanleihen	D_Corp, s_c, default prob., LGD, position size	Bond-Modell Inkl. Credit Spread und Loss Given Default
3	Equities	Aktien	μ_3, σ_3 , Freiheitsgrade, mean reversion	Log-Normal Rendite mit optionalem Fat Tail und optionaler mean reversion
4	RealEstate	Immobilien	μ_4, σ_4 , Freiheitsgrade	Log-Normal Rendite mit optionalem Fat Tail
5	Alternatives	Infrastruktur Debt	D_Alt, s_a, default prob., LGD, position size	Bond-Modell (Infrastructure Debt)

Korrelationsmatrix

Die Abhängigkeitsstruktur wird nur noch zwischen den drei stochastisch frei modellierten Faktoren — Basiszins (InterestRate), Aktien (Equities) und Immobilien (RealEstate) — über die Korrelationsmatrix Σ abgebildet. Technisch ist Σ im Code als 6×6-Matrix über alle sechs Faktoren implementiert; die für die drei Anleihenklassen gezogenen Zufallsrenditen werden jedoch verworfen und vollständig durch das Bond-Modell ersetzt, sodass ökonomisch nur die 3×3-Substruktur (Zins, Aktien, Immobilien) wirksam ist. Die Renditen der Anleihen (Staats-, Unternehmens- und Infrastrukturanleihen) werden nicht über die Korrelationsmatrix erzeugt, sondern vollständig aus dem Bond-Modell (Coupon, Duration-Effekt, Pull-to-Par, Default) abgeleitet. Ihre Abhängigkeit von den übrigen Märkten entsteht damit indirekt über den gemeinsamen Basiszins $r1_t$. Die Cholesky-Zerlegung L der Matrix wird vorab berechnet, sodass $\Sigma = L \cdot L^T$:

$$\Sigma = L \cdot L^T, \quad \text{wobei } L = \text{cholesky}(\Sigma) \quad (\text{Cholesky})$$

3. Simulation korrelierter Renditen

Funktion: `simulate_all_returns()`

Die Renditesimulation erfolgt in mehreren Schritten, um korrelierte Fat-Tail-Verteilungen zu erzeugen. Der Ansatz basiert auf einer Gaussian Copula mit optionaler Student-t Marginaltransformation.

Schritt 1: Korrelierte Standardnormalverteilungen

Für jeden Zeitpunkt $t \in \{1, \dots, T\}$ wird ein Vektor Z_t aus unabhängigen Standard-Normalverteilungen für alle sechs Faktoren erzeugt und mittels der Cholesky-Matrix L korreliert; wirksam sind davon nur die drei freien Faktoren (Zins, Aktien, Immobilien), da die Anleihen-Spalten anschliessend verworfen werden. Die Bond-Renditen werden anschliessend separat im Bond-Modell berechnet:

$$Z_t \sim N(0, I_n), \quad Z_t \in \mathbb{R}^6$$
$$X_t = Z_t \cdot L^T, \quad X_t \sim N(0, \Sigma)$$

Schritt 2: Transformation zu Fat Tails (Gaussian Copula)

Falls für eine Anlageklasse i Freiheitsgrade $v_i < 100$ definiert sind, wird die korrelierte Normalverteilung über den Copula-Ansatz in eine Student-t-Verteilung transformiert:

$$u_i = \Phi(X_{t,i}) \quad \rightarrow \text{Transformation zu Uniform}(0,1)$$
$$X'_{t,i} = t^{-1}_{v_i}(u_i) \cdot \sqrt{[(v_i - 2) / v_i]} \quad \rightarrow \text{Inverse CDF der t-Verteilung}$$

Der Skalierungsfaktor $\sqrt{[(v-2)/v]}$ stellt sicher, dass die transformierte Variable Varianz 1 hat. Kleinere v -Werte erzeugen dickere Tails (mehr Extremereignisse).

Schritt 3: Log-Renditen mit Drift-Korrektur

Aus den korrelierten (ggf. fat-tailed) Zufallsvariablen werden die Log-Renditen berechnet:

$$\ln(1 + r_{t,i}) = (\mu_i - \sigma_i^2/2) + \sigma_i \cdot X'_{t,i}$$
$$r_{t,i} = \exp[\ln(1 + r_{t,i})] - 1$$

Die Drift-Korrektur $-\sigma^2/2$ (Jensen's Inequality Correction) stellt für normalverteilte Faktoren sicher, dass der erwartete Drift μ_i korrekt getroffen wird; für Student-t-transformierte Faktoren gilt die Korrektur nur näherungsweise.

4. Bond-Renditemodell

Das Modell verwendet ein realistisches Bond-Modell für Staatsanleihen (Gov), Unternehmensanleihen (Corp) und Infrastructure Debt (Alt). Die Gesamtrendite setzt sich aus vier Komponenten zusammen.

Die Bond-Renditen werden ausschliesslich über dieses Modell bestimmt und nicht aus der Korrelationsmatrix gezogen. Der gemeinsame Treiber aller Anleihen ist der simulierte Basiszins $r1_t$, wodurch eine indirekte Korrelation zu den übrigen Anlageklassen entsteht.

4.1 Zinsstrukturkurve

Der Marktzins für eine Anlageklasse wird aus dem simulierten Basiszins $r1_t$, der aktuellen Duration D und der Steigung der Zinsstrukturkurve (Slope) hergeleitet:

$$i_Gov(t) = r1_t + D_Gov(t) \cdot Slope \quad (\text{Gov Bonds Zins})$$

$$i_Corp(t) = r1_t + D_Corp(t) \cdot Slope + s_c \quad (\text{Corp Bonds Zins})$$

$$i_Alt(t) = r1_t + D_Alt(t) \cdot Slope + s_a \quad (\text{Infrastructure Debt Zins})$$

Wobei s_c der Credit Spread für Unternehmensanleihen und s_a der Spread für Infrastructure Debt ist.

4.2 Coupon-Rendite

Der Coupon wird bei Kauf fixiert und bleibt bis zur nächsten Neuanlage konstant. Die effektive Coupon-Rendite bezieht sich auf den aktuellen Marktwert (MV), nicht den Par-Wert:

$$r_Coupon(t) = C_fix / MV(t) \quad (\text{Coupon-Rendite})$$

Bei einem Discount-Bond ($MV < 1$) ist die effektive Coupon-Rendite höher als der nominale Coupon, bei einem Premium-Bond ($MV > 1$) entsprechend niedriger.

4.3 Duration-Effekt (Kursänderung durch Zinsänderung)

Die Kursänderung aufgrund von Zinsänderungen wird mittels modifizierter Duration und Konvexität berechnet. Es wird eine Taylor-Approximation 2. Ordnung verwendet:

$$\Delta r1 = r1_t - r1_{\{t-1\}} \quad (\text{Zinsänderung})$$

$$Konvexität = (D^2 + D) / (1 + i)^2$$

$$Duration\text{-Komponente} = -D \cdot \Delta r1 / (1 + i)$$

$$Konvexitäts\text{-Komponente} = 0.5 \cdot Konvexität \cdot (\Delta r1)^2$$

$$r_Duration(t) = Duration\text{-Komponente} + Konvexitäts\text{-Komponente}$$

Der Duration-Effekt wird auf [-50%, +50%] begrenzt, um unrealistische Kurssprünge zu vermeiden. Bei steigenden Zinsen ($\Delta r1 > 0$) sinkt der Bondkurs (negativer Duration-Effekt), bei sinkenden Zinsen steigt er.

4.4 Pull-to-Par-Effekt

Der Bond konvergiert linear gegen Par (100%) bei Fälligkeit. Der jährliche Amortisationsbetrag wird additiv zum Marktwert addiert:

$$PtP_absolut = (1 - MV) / D_Rest \quad (\text{Pull-to-Par})$$

Bei einem Discount ($MV < 1$) ist PtP positiv (Wertsteigerung), bei einem Premium ($MV > 1$) negativ (Wertverlust). Der Effekt wird auf [-10%, +10%] pro Jahr begrenzt.

4.5 Marktwert-Update

Der Marktwert wird in jedem Zeitschritt aktualisiert. Der Duration-Effekt wirkt multiplikativ, Pull-to-Par additiv:

$$MV_{nach_Duration} = MV(t) \cdot (1 + r_{Duration})$$

$$MV(t+1) = MV_{nach_Duration} + PtP_{absolut}$$

$$MV(t+1) \in [0.5, 1.5] \quad (Begrenzung)$$

4.6 Default-Verlust (nur Corp Bonds & Infrastructure Debt)

Corporate Bonds und Infrastructure Debt haben ein stochastisches Default-Risiko:

$$\text{Falls } U \sim \text{Uniform}(0,1) < p_{Default} :$$

$$L_{Default} = Exposure \cdot LGD$$

$$\text{Sonst: } L_{Default} = 0$$

Variable	Typ	Beschreibung
$p_{Default}$	float	Jährliche Ausfallwahrscheinlichkeit (z.B. 0.3% für Corp, 1.3% für Infra)
$Exposure$	float	Anteil des Portfolios, der bei Default betroffen ist (z.B. 2%)
LGD	float	Loss Given Default - Verlustquote bei Ausfall (z.B. 40%)

4.7 Gesamte Bond-Rendite

$$r_{Bond}(t) = r_{Coupon}(t) + (MV(t+1) - MV(t)) / MV(t) - L_{Default} \quad (\text{Total Bond Return})$$

Die Gesamtrendite setzt sich zusammen aus: Coupon-Einkommen + Marktwertänderung (Duration-Effekt + Pull-to-Par) - eventuelle Default-Verluste.

5. Sterblichkeitsmodellierung und Bestandsführung

5.1 Sterbetafeln (BFS-Referenzszenario)

Die Sterblichkeit wird basierend auf Schweizer Sterbetafeln (Periodentafel, BFS-Referenzszenario) modelliert; künftige Sterblichkeitsverbesserungen (Generationeneffekte) werden nicht explizit abgebildet. Für jedes Alter x und Geschlecht g wird die einjährige Sterbewahrscheinlichkeit q_x aus der Tafel gelesen:

$$q_{x,g} = P(\text{Tod im Alter } x \mid \text{Geschlecht } g) \quad (\text{Sterbewahrscheinlichkeit})$$

Funktion: `get_qx(age, gender, survival_table)`

Für Alter > 110 oder < 0 wird $q_x = 1.0$ (sicherer Tod) angenommen.

5.2 Vorberechnete Sterblichkeits-Arrays

Funktion: `precompute_mortality_arrays()` — Für Performance werden die q_x -Werte in NumPy-Arrays vorberechnet (`qx_M[age]` und `qx_F[age]`).

5.3 Kumulative Überlebenswahrscheinlichkeiten

Funktion: `precompute_survival_probs()`

Die kumulative Wahrscheinlichkeit, dass eine Person mit Startalter x noch k Jahre überlebt:

$${}_k p_x = \prod_{j=0}^{k-1} (1 - q_{x+j})$$
$${}_0 p_x = 1 \quad (\text{Startwert})$$

5.4 Timing-Konvention und Bestandsführung

Die Simulation folgt dieser Reihenfolge pro Zeitschritt t :

- 1. Rentenzahlung: Alle lebenden Rentner erhalten ihre Rente (nachschüssig)
- 2. Sterblichkeitsprüfung: Für jede Person wird mit Wahrscheinlichkeit q_x der Tod simuliert
- 3. Witwenrente: Bei Tod eines Verheirateten wird der überlebende Partner als Witwe/r aufgenommen
- 4. Alterung: Alle Überlebenden werden um 1 Jahr gealtert

5.5 Ehegattenrente (Witwenrente)

Stirbt ein verheirateter Rentner, erhält der überlebende Ehepartner eine Witwenrente.

Modellhinweis: Ein Vorversterben des Ehepartners wird in der Simulation nicht abgebildet – beim Tod eines verheirateten Rentners wird stets eine Hinterbliebenenrente ausgelöst, während die Bewertung $W(t)$ die Anwartschaft mit der Überlebenswahrscheinlichkeit des Partners gewichtet (Kap. 6.4). Die Simulation zahlt damit tendenziell häufiger Hinterbliebenenrenten aus, als reserviert ist; die ausgewiesenen Unterdeckungswahrscheinlichkeiten sind insoweit konservativ:

$$R_{\text{Witwe}} = R_{\text{initial}} \cdot \alpha_{\text{spouse}} \quad (\text{Witwenrente})$$

Variable	Typ	Beschreibung
R_{initial}	CHF	Ursprüngliche Rente des Verstorbenen (InitialPension)
α_{spouse}	float	Ehegattenrente-Satz (Default: 40% = 0.40)
SpouseAgeDiff	int	Altersdifferenz zum Partner (Default: M -3, F +3)

6. Barwertberechnung der Verpflichtungen $W(t)$

Funktion: `calculate_liability_barwert_base()`

6.1 Technischer Zinssatz

Der technische Zinssatz i_{tech} wird aus dem Basiszins, der Zinsstruktur und einem Liability-Discount-Spread hergeleitet:

$$i_{tech}(t) = \max(r1_t + D_tech \cdot Slope + Spread_L, Floor) \quad (\text{Tech. Zins})$$

Variable	Typ	Beschreibung
D_tech	float	Duration für technischen Zinssatz (TECHNICAL_RATE_DURATION)
$Slope$	float	Steigung der Zinsstrukturkurve (YIELD_CURVE_SLOPE)
$Spread_L$	float	Liability Discount Spread (LIABILITY_DISCOUNT_SPREAD)
$Floor$	float	Mindest-Technischer-Zinssatz (TECHNICAL_RATE_FLOOR)

6.2 Diskontfaktor

$$v = 1 / (1 + i_{tech}) \quad (\text{Diskontfaktor})$$

6.3 Barwert der eigenen Rente (Leibrentenbarwert)

Funktion: `precompute_annuity_factors()`

Der Leibrentenbarwertfaktor a_x gibt den Barwert einer jährlichen Zahlung von 1 CHF an eine Person im Alter x an, solange diese lebt (nachsüssige Zahlweise; daher die aktuarielle Notation a_x anstelle des vorschüssigen $\ddot{a}_x$). Die Berechnung erfolgt rekursiv von oben nach unten:

$$a_x = v \cdot p_x \cdot (1 + a_{x+1})$$
$$a_{\{max_age\}} = 0 \quad (\text{Randbedingung})$$

Ausgeschrieben:

$$a_x = \sum_{k=1}^{\omega-x} \{\omega-x\} v^k \cdot {}_k p_x \quad (\text{Leibrentenbarwert})$$

wobei $\omega = 110$ (maximales Alter) und ${}_k p_x$ die kumulative Überlebenswahrscheinlichkeit ist.

Der Barwert der eigenen Rente für Person i :

$$BW_eigene(i) = R_i \cdot a_{\{x_i, g_i\}} \quad (\text{BW eigene Rente})$$

wobei R_i die jährliche Rente und x_i, g_i Alter und Geschlecht der Person sind.

6.4 Barwert der Ehegattenanwartschaft

Für verheiratete Rentner wird die Anwartschaft auf eine zukünftige Ehegattenrente berechnet. Dies erfordert eine doppelte Summation über alle möglichen Todeszeitpunkte des Rentners und die bedingte Überlebenswahrscheinlichkeit des Ehepartners. Hinweis: Durch die Gewichtung mit ${}_{k+1}p_y$ in Kombination mit $a_{\{y+k\}}$ ist das Überlebensjahr $k \rightarrow k+1$ des Partners doppelt enthalten; die Anwartschaft wird dadurch geringfügig unterschätzt. In den Cashflow- und Duration-Funktionen (Kap. 8.2 und 9.2) wird das Überleben des Partners bis zum Tod des Rentners – konsistent mit der Simulationsdynamik (Kap. 5.4/5.5) – nicht gewichtet:

$$BW_Anwartschaft(i) = R_spouse \cdot \sum_{k=0}^{\omega-x} \{\omega-x\} [$$

$$\begin{aligned}
 &P(\text{Rentner stirbt in Jahr } k) \\
 &\cdot P(\text{Ehepartner lebt in Jahr } k) \\
 &\cdot v^k \cdot a_{\{y+k, g_{\text{spouse}}\}} \\
 &]
 \end{aligned}$$

Wobei:

$$\begin{aligned}
 P(\text{Rentner stirbt in } k) &= {}_k p_x \cdot q_{x+k} \\
 P(\text{Ehepartner lebt in } k) &= {}_{k+1} p_y \\
 y &= x + \text{SpouseAgeDiff} \quad (\text{Alter des Ehepartners})
 \end{aligned}$$

6.5 Gesamte Verpflichtung $W(t)$

$$W(t) = \sum_{i \in \text{Lebende}} [BW_{\text{eigene}}(i) + BW_{\text{Anwartschaft}}(i)] \quad (\text{Gesamt-Liability})$$

Die Summe läuft über alle lebenden Personen im Bestand (Status $\neq$ Dead), einschliesslich der bereits verwitweten Personen (die keine eigene Anwartschaft mehr generieren, aber deren eigene Rente im Barwert enthalten ist).

7. Leibrentenbarwertfaktor — Detailberechnung

7.1 Rekursive Berechnung

Die Berechnung erfolgt rückwärts vom Maximalalter $\omega = 110$ zum Alter 0. Die Verwaltungskosten (Admin Fee) pro Person werden wie eine zusätzliche jährliche Zahlung behandelt und mit demselben Leibrentenbarwertfaktor kapitalisiert, d.h. $BW = (R + \text{Fee}) \cdot a_x$:

$$a_\omega = 0$$
$$a_x = v \cdot (1 - q_x) \cdot (1 + a_{x+1}) \quad \text{für } x = \omega-1, \omega-2, \dots, 0$$

Diese rekursive Formel ist äquivalent zur direkten Summenformel, aber effizienter zu berechnen ($O(n)$ statt $O(n^2)$).

7.2 Cache-Mechanismus

Die Annuity-Faktoren werden nur neu berechnet, wenn sich der technische Zinssatz signifikant ändert ($|i_{\text{tech_neu}} - i_{\text{tech_cached}}| > 0.0001$). Dies vermeidet redundante Berechnungen und beschleunigt die Simulation erheblich.

8. Duration-Berechnung (Macaulay-Duration der Liabilities)

Funktion: `calculate_liability_duration()`

8.1 Macaulay-Duration — Definition

Die Macaulay-Duration misst die gewichtete mittlere Laufzeit aller zukünftigen Cashflows. Für die Liabilities einer Pensionskasse ist dies der Zeitpunkt, zu dem im Durchschnitt die Verpflichtungen fällig werden.

$$D_{Mac} = \sum (k \cdot CF_k \cdot v^k) / \sum (CF_k \cdot v^k) \quad (\text{Macaulay-Duration})$$

8.2 Duration pro Versicherten

Für jede Person i wird die individuelle Duration berechnet, die sowohl die eigene Rente als auch die Ehegattenanwartschaft berücksichtigt:

Eigene Rente:

$$PV_{eigene}(i) = \sum_{k=1}^{\omega-x} R_i \cdot {}_k p_x \cdot v^k$$
$$WT_{eigene}(i) = \sum_{k=1}^{\omega-x} k \cdot R_i \cdot {}_k p_x \cdot v^k$$

Ehegattenanwartschaft (für verheiratete Aktive):

$$PV_{spouse}(i) = \sum_k \sum_j R_{sp} \cdot P(\text{Tod in } k) \cdot P(\text{Witwe lebt } j) \cdot v^{k+j}$$
$$WT_{spouse}(i) = \sum_k \sum_j (k+j) \cdot R_{sp} \cdot P(\text{Tod in } k) \cdot P(\text{Witwe lebt } j) \cdot v^{k+j}$$

Individuelle Duration:

$$D_i = (WT_{eigene}(i) + WT_{spouse}(i)) / (PV_{eigene}(i) + PV_{spouse}(i)) \quad (\text{Duration Person } i)$$

8.3 Portfolio-gewichtete Gesamt-Duration

$$D_{Liab} = \sum (PV_i \cdot D_i) / \sum (PV_i) \quad (\text{Gesamt-Duration})$$

Die Gesamt-Duration wird barwertgewichtet berechnet: Personen mit höheren Verpflichtungen haben einen grösseren Einfluss auf die Gesamt-Duration.

8.4 Berechnungsintervall

Die Liability-Duration wird standardmässig alle 5 Jahre exakt berechnet. In den Zwischenjahren wird sie um 1 pro Jahr approximiert (Duration sinkt, da der Bestand altert). Im Liability-Matching-Modus wird sie jährlich berechnet.

9. Duration-Management und Cash Flow Matching

Funktion: `get_duration_for_mode()`

9.1 Duration-Modi

Modus	Verhalten	Formel
fixed	Duration sinkt jährlich um 1	$D(t+1) = \max(0, D(t) - 1)$
fixed_reset	Wie fixed, aber periodischer Reset	$D(t+1) = D_{init}$ wenn $(t+1) \bmod \text{Interval} = 0$
liability_matching	Duration = Liability-Duration (jährlich)	$D(t) = D_{Liab}(t)$
cashflow_matching	Gewichtete Duration der CFM-Tranchen	$D(t) = \sum (D_k \cdot w_k) / \sum (w_k)$

9.2 Cash Flow Matching (CFM)

Funktion: `calculate_expected_cashflows()`, `calculate_cfm_tranches()`

Beim Cash Flow Matching werden die erwarteten zukünftigen Rentenzahlungen berechnet und mit Bond-Tranchen dedizierter Laufzeiten abgedeckt:

- Schritt 1: Erwartete Cashflows berechnen

$$E[CF_k] = \sum_i (R_i + Fee) \cdot p_{\{X_i\}} + E[CF_{spouse,k}] \quad (\text{Erwartete Cashflows})$$

- Schritt 2: Barwerte der Cashflows

$$PV_k = E[CF_k] / (1 + d)^k \quad (\text{Barwert Cashflow})$$

wobei $d = r1_{erwartet} + \text{Slope} \cdot 10$ (mindestens 1%).

- Schritt 3: Gewichtung der Tranchen

$$w_k = PV_k / \sum PV_j \quad (\text{Tranchen-Gewichte})$$

- Schritt 4: Bond-Beträge pro Tranche

$$V_k = w_k \cdot V_{Bond_Portfolio} \quad (\text{Tranchen-Beträge})$$

9.3 CFM-Coupon-Berechnung

Funktion: `get_cfm_coupon_rate()`

Der gewichtete durchschnittliche Coupon des CFM-Portfolios basiert auf den bei Kauf ($t=0$) fixierten Coupons der einzelnen Tranchen:

$$C_{CFM}(t) = \sum_{\{k>t\}} [C_k \cdot w_k] / \sum_{\{k>t\}} w_k \quad (\text{CFM-Coupon})$$

wobei $C_k = r1_{initial} + D_k \cdot \text{Slope} + \text{Spread}$ der Coupon der Tranche k ist.

10. Vermögensentwicklung und Deckungsgrad

Funktion: `run_monte_carlo_path_full()` — Abschnitt D

10.1 Portfolio-Rendite

Die Gesamtrendite des Portfolios ist die gewichtete Summe der Einzelrenditen:

$$r_{port}(t) = w_{Gov} \cdot r_{Gov}(t) + w_{Corp} \cdot r_{Corp}(t) + w_{Eq} \cdot r_{Eq}(t) + w_{RE} \cdot r_{RE}(t) + w_{Alt} \cdot r_{Alt}(t) \quad (\text{Portfolio-Rendite})$$

Hinweis: $w_{InterestRate} = 0$ (kein direktes Investment in den Basiszins).

10.2 Mean Reversion für Aktien (optional)

Bei aktivierter Mean Reversion wird nach starken kumulierten Verlusten ein zusätzlicher positiver Drift hinzugefügt:

$$CumReturn_{Eq}(t) = \prod_{s=1}^t (1 + r_{Eq}(s)) - 1$$

$$ExpReturn_{Eq}(t) = (1 + \mu_{LT})^t - 1$$

$$Shortfall(t) = CumReturn_{Eq}(t) - ExpReturn_{Eq}(t)$$

Falls $Shortfall < Threshold$:

$$Adj = \min(-Strength \cdot Shortfall, 0.15)$$

$$r'_{Eq}(t) = r_{Eq}(t) + Adj$$

10.3 Vermögensgleichung

$$V(t) = V(t-1) \cdot (1 + r_{port}(t)) - CF_{Total}(t) + Cap_{Payout}(t) \quad (\text{Vermögen})$$

Variable	Typ	Beschreibung
$V(t)$	CHF	Vermögen (Assets) am Ende von Jahr t
$r_{port}(t)$	float	Gewichtete Portfoliorendite in Jahr t
$CF_{Total}(t)$	CHF	Gesamte Cashflows = Renten + Verwaltungskosten + Sonder-Rente
$Cap_{Payout}(t)$	CHF	Auszahlung aus Interest Rate Cap (falls aktiv)

10.4 Deckungsgrad (Funding Ratio)

$$DG(t) = V(t) / W(t) \quad \text{falls } W(t) > 0$$

$$DG(t) = 2.0 \quad \text{falls } W(t) = 0 \text{ und } V(t) > 0$$

$$DG(t) = 0.0 \quad \text{sonst}$$

Der Deckungsgrad ist die zentrale Kennzahl der Simulation. Ein $DG < 100\%$ bedeutet Unterdeckung im Sinne von Art. 44 Abs. 1 BVV 2; ein $DG < 80\%$ gilt im Modell als kritische Unterdeckung und bildet die Schwelle der Risikometrik $P(\text{kritische Unterdeckung})$ (Kap. 12.1). Als Default gilt demgegenüber die vollständige Erschöpfung des Vermögens ($V \leq 0$).

10.5 Initialer Barwert und Startkapital

Funktion: `calculate_liability_barwert_initial()`

$$i_{tech,0} = r1_{erwartet} + D_{tech} \cdot Slope + Spread_L$$

$$W(0) = Barwert(Bestand, i_{tech,0})$$

$$V(0) = W(0) \cdot (1 + GRR)$$

Variable	Typ	Beschreibung
----------	-----	--------------

<i>GRR</i>	float	General Reserve Rate (Wertschwankungsreserve)
<i>DG(0)</i>	float	= 1 + GRR (Start-Deckungsgrad, z.B. 1.05 = 105%)

11. Spezialfunktionen

11.1 Interest Rate Cap

Ein Interest Rate Cap schützt gegen steigende Zinsen. Die Prämie wird am Anfang bezahlt und reduziert das Startkapital:

$$Notional = V(0) \cdot (w_Gov + w_Corp)$$

$$Prämie = Notional \cdot Premium_Rate$$

$$V(0)_{eff} = V(0) - Prämie$$

Falls $r1_t > r1_0 + Strike$ und $t < Cap_Duration$:

$$Payout = (r1_t - Strike_Level) \cdot V(t-1) \cdot (w_Gov + w_Corp)$$

11.2 Sonder-Rente (Special Pension Payout)

Bei Überdeckung über einem Schwellwert wird eine einmalige Sonder-Rente ausgeschüttet, um den Deckungsgrad auf ein Ziel-Niveau zu senken:

Falls $DG(t) > DG_Threshold$:

$$V_target = W(t) \cdot DG_Target$$

$$Payout = \max(0, V(t) - V_target)$$

$$V(t)_{neu} = V(t) - Payout$$

11.3 Sammelstiftung-Modus (Modellierung einer Gemeinschaftseinrichtung)

Der Codeparameter «sammelstiftung_*» ist historisch bedingt: Strukturell modelliert der Modus eine Gemeinschaftseinrichtung (gemeinsamer Deckungsgrad und kollektive Anlage über alle Kohorten), nicht eine Sammeleinrichtung mit getrennten Vorsorgewerken. Im Sammelstiftung-Modus wird alle X Jahre ein neuer Rentnerbestand (identisch zum initialen Bestand) hinzugefügt, sofern der Deckungsgrad > 100%:

Falls $(t+1) \bmod Interval = 0$ und $V(t) > W(t)$:

$$W_neu = Barwert(neuer\ Bestand, i_tech)$$

$$V_neu = W_neu \cdot (1 + GRR)$$

$$W(t) = W(t) + W_neu$$

$$V(t) = V(t) + V_neu$$

12. Risikometriken und Optimierung

12.1 Risikometriken

Funktion: calculate_risk_metrics()

$$P(\text{kritische Unterdeckung}) = |\{\text{Pfade mit } \min(DG) < 80\% \}| / N_{\text{Pfade}}$$

$$P(\text{Default}) = |\{\text{Pfade mit } \min(V_t) \leq 0\}| / N_{\text{Pfade}}$$

«Kritische Unterdeckung» bezeichnet die modellinterne 80%-Schwelle; eine Unterdeckung im Sinne von Art. 44 Abs. 1 BVV 2 liegt bereits bei $DG < 100\%$ vor. Für jeden Event werden die Ursachen analysiert (Aktiencrash, Zinssenkung, Zinsanstieg, Corporate Defaults, Liability-Explosion).

12.2 Zielfunktion der Optimierung

Datei: optimize_alm_beta.py

Die Optimierung minimiert eine gewichtete Kombination der Risikometriken:

$$\text{Objective} = \alpha \cdot P(\text{kritische Unterdeckung}) + \beta \cdot P(\text{Default}) + \gamma \cdot E[\text{Shortfall}] \quad (\text{Zielfunktion})$$

Variable	Typ	Beschreibung
α	float	Gewicht für Unterdeckung (Default: 1.0)
β	float	Gewicht für Default (Default: 5.0)
γ	float	Gewicht für erwarteten Shortfall (Default: 0.00001)

12.3 Optimierte Parameter

- Asset-Gewichte: w_{Gov} , w_{Corp} , w_{Eq} , w_{RE} (w_{Alt} als Residuum)
- Bond-Durations: $D_{\text{Gov_initial}}$, $D_{\text{Corp_initial}}$
- Nebenbedingungen: $\sum w_i = 1$, $w_i \geq 0$, $w_{\text{Alt}} \leq 25\%$ (modellinterne Obergrenze; reglementarisch gelten 10% für Infrastrukturanlagen nach Art. 53 Abs. 1 lit. dbis i.V.m. Art. 55 lit. f BVV 2 bzw. 15% für alternative Anlagen nach Art. 55 lit. e BVV 2; Überschreitungen bedürfen der Erweiterungsbegründung nach Art. 50 Abs. 4 BVV 2)

13. Systematische Sensitivitätsanalyse (sensitivity_analysis.py)

Das Skript `sensitivity_analysis.py` erzeugt und simuliert systematisch eine grosse Anzahl von Szenarien, um den Einfluss von Demografie, Asset-Allokation, Duration-Strategie und Sammelstiftungs-Modus auf die Risikokennzahlen zu untersuchen. Jedes Szenario wird als eigene CSV-Datei (`sensitivity_scenario_XXXX.csv`) exportiert und anschliessend in R ausgewertet.

13.1 Variierte Dimensionen

Vier Parameterräume werden kombiniert (kartesisches Produkt aus Populations-Varianten und Strategie-Kombinationen):

- Rentnerbestand (Demografie): Bestandesgrösse, Durchschnittsalter, Rentenhöhe, Anteil Verheiratete, Ehegattenrente, Altersdifferenz, Frauenanteil.
- Asset-Allokation: aus Bandbreiten (BVV2-konform) werden via Latin Hypercube Sampling (LHS) gültige Gewichtskombinationen mit Summe = 1.0 erzeugt.
- Duration-Strategien: kartesisches Produkt der Modi (gov/corp/alt) mal LHS-Stichproben der initialen Durationen.
- Sammelstiftung (Gemeinschaftseinrichtungs-Modus, vgl. Kap. 11.3): aktiviert/deaktiviert mit konfigurierbarem Intervall.

13.2 Funktionsübersicht

Die folgende Tabelle fasst die zentralen Funktionen des Skripts zusammen.

Funktion	Zweck	Outputs
<code>_lhs_sample()</code>	Latin Hypercube Sampling über diskrete Parameter-Grids. Jede Dimension wird in <code>n_samples</code> gleich grosse Schichten (Strata) eingeteilt; pro Stratum wird ein Wert gezogen und die Dimensionen unabhängig permutiert (maximale Spreizung).	Liste von Parameter-Dicts (Populations-Varianten)
<code>_lhs_allocation_sample()</code>	Erzeugt gültige Asset-Allokationen via LHS in $[0,1]^5$, skaliert auf die Bandbreiten, normiert auf Summe = 1.0 und behält per Rejection-Sampling nur bandbreitenkonforme Gewichte.	Liste von Gewichts-Tuples (gov, corp, eq, re, alt)
<code>_lhs_duration_sample()</code>	Bildet das kartesische Produkt der Duration-Modi (gov/corp/alt) und kombiniert es mit LHS-Stichproben der kontinuierlichen Init-Duration-Bandbreiten.	Liste von Tuples (Modi + initiale Durationen)
<code>build_scenario_list()</code>	Baut die vollständige Szenario-Liste: LHS-Populationsraum (inkl. Referenz-Basisszenario) gekreuzt mit Allokationen, Duration-Strategien und Sammelstiftung. Vergibt fortlaufende <code>scenario_id</code> .	Liste von Szenario-Dicts
<code>generate_population_df()</code>	Generiert einen Rentnerbestand aus den Populations-Parametern (Alter, Geschlecht, Ehestand, Rentenhöhe ~ Normalverteilung mit ca. 35% CV, Ehegatten-Altersdifferenz, Ehegattenrenten-Satz).	<code>pandas.DataFrame</code> (Rentnerbestand)
<code>apply_scenario_to_config()</code>	Überschreibt die relevanten <code>cfg</code> -Attribute (WEIGHTS, Duration-Modi/-Durationen, Sammelstiftung) für ein einzelnes Szenario vor dem Simulationslauf.	— (mutiert <code>cfg</code> in place)
<code>run_single_scenario()</code>	Führt ein Szenario vollständig aus: Config anpassen (<code>apply_scenario_to_config</code>), Population vorbereiten, <code>W(0)/V(0)</code> berechnen, die parallele Monte-Carlo-Simulation starten, Risikometriken berechnen und die Pfade via <code>export_scenario_csv</code> exportieren.	Dict mit Risikometriken und Metadaten
<code>export_scenario_csv()</code>	Exportiert alle MC-Pfade eines Szenarios als CSV (Format analog zu <code>export_all_paths_csv</code> aus <code>main_alm_simulation.py</code>), erweitert um	Pfad zur exportierten Szenario-CSV

	scenario_id und sämtliche Szenario- und Populations-Parameter als konstante Spalten.	
check_disk_space()	Prüft freien Speicherplatz und bisher belegte Datenmenge im Ausgabeverzeichnis; signalisiert Abbruch bei Unterschreiten der Limits (min_free_gb / max_disk_gb).	(ok, free_gb, used_gb)
format_duration()	Formatiert eine Dauer in Sekunden in ein lesbares Format (s / min / h min) für die Fortschrittsanzeige.	String
load_scenarios_from_file()	Lädt eine benutzerdefinierte Szenario-Liste aus einer JSON-Datei (Schlüssel 'scenarios' oder direkte Liste).	Liste von Szenario-Dicts
_signal_handler()	Graceful-Shutdown-Handler für Ctrl+C / SIGTERM: schliesst das laufende Szenario ab; bei erneutem Signal sofortiger Abbruch.	— (setzt Abbruch-Flag)
main()	Einstiegspunkt: parst die Kommandozeilenargumente (n_paths, t_horizon, LHS-Stichproben/Seeds, Speicherlimits, resume, dry_run), baut die Szenarien und steuert die parallele Simulations-Schleife inkl. Resume- und Speicherplatz-Logik.	— (Skript-Steuerung, CSV-Exporte)

13.3 Wichtige Kommandozeilen-Optionen

Steuerung erfolgt über Argumente, u.a.: --n_paths (MC-Pfade je Szenario), --t_horizon, --output_dir, --lhs_samples (Populations-Varianten), --alloc_samples (Allokationen), --dur_samples (Duration-Stichproben), --resume (ab Szenario fortsetzen), --dry_run (nur Szenarien anzeigen), --no_path_export (nur Summary), --max_disk_gb / --min_free_gb (Speicherlimits) und --config (eigene JSON-Szenarien).

13.4 Generierung von Zufallsbeständen (rentnerbestand_generator.py)

Das Hilfsskript rentnerbestand_generator.py erzeugt einen synthetischen Rentnerbestand auf Basis weniger Parameter und speichert ihn als data/rentnerbestand_initial.csv. Pro Person werden Alter (Normalverteilung um einen Mittelwert, begrenzt auf 65–100), Geschlecht (konfigurierbarer Frauenanteil), Zivilstand (verheiratet/ledig/verwitwet), Rentenhöhe (InitialPension, normalverteilt und begrenzt), Ehegatten-Altersdifferenz (geschlechtsabhängig, negativ = Partner jünger), Ehegattenrenten-Satz (SpousePensionRate) sowie Geburtsjahr erzeugt.

Das Skript dient unter anderem der automatisierten Analyse der Cashflow-Sensitivität in Abhängigkeit der Eigenschaften der Rentnerkohorte (z.B. Bestandsgrösse, Durchschnittsalter, Frauenanteil, Anteil Verheiratete, Rentenhöhe und -streuung, Ehegattenrente). In der systematischen Sensitivitätsanalyse (Kap. 13.1–13.2) übernimmt die Funktion generate_population_df() dieselbe Aufgabe parametrisiert pro Szenario.

14. R-Analysefunktionen (functions.R / functions_parallel.R) — Kurzübersicht

Die R-Funktionsdatei stellt Analyse- und Visualisierungsfunktionen bereit, die auf den exportierten CSV-Daten der Monte-Carlo-Simulation arbeiten.

Funktion	Zweck	Outputs
load_mc_data()	Lädt alle CSV-Dateien aus dem Datenordner und kombiniert sie zu einem DataFrame. Erstellt eindeutige Pfad-Nummern über mehrere Dateien hinweg.	data.frame mit allen Simulationspfaden
analyze_default_by_interest_shock()	Analysiert den Zusammenhang zwischen Zinsschocks und Default-	Barplot Default-Rate, Density-Plot r1_t, Dezil-Analyse, Chi²-Test

	Wahrscheinlichkeit. Berechnet relatives Risiko und führt Chi ² -Tests durch.	
<code>analyze_r1t_default_comparison()</code>	Vergleicht die <code>r1_t</code> -Verteilung zwischen Default-Pfaden und allen Pfaden. Nutzt Kolmogorov-Smirnov-Tests zur Signifikanzprüfung.	Density-Plot, Box-Plot, KS-Test-Ergebnisse
<code>plot_path_analysis()</code>	Erstellt einen Einzelpfad-Plot mit Deckungsgrad-Zeitreihe und Referenzlinien (80%, 100%).	ggplot-Objekt
<code>plot_boxplot_difference()</code>	Box-Plot des Deckungsgrads für ein bestimmtes Jahr mit optionalen Datenpunkten.	ggplot-Objekt
<code>plot_path_analysis_detailed()</code>	Detaillierte Mehr-Panel-Analyse eines Pfades inkl. Witwen, <code>V_t</code> , <code>W_t</code> , Rentner-Anzahl mit Vergleich zu statistischen Benchmarks (Median, P5-P95).	4-Panel-Plot (Witwen/Returns, <code>V_t/W_t/DG</code> , Rentner, Parameter-Tabelle)
<code>analyze_cashflow_dispersion()</code>	Analysiert die Streuung der Cashflows in Abhängigkeit der Bestandseigenschaften (Grösse, Alter, Geschlecht, Ehestand, Pensions-Heterogenität).	10 Plots inkl. Heatmap und Korrelationsmatrix
<code>analyze_return_volatility()</code>	Analysiert die Portfolio-Return-Volatilität nach Asset Allocation und Duration. Unterstützt Duration-Matching (Neutralisierung des Duration-Effekts).	Efficient Frontier, Ranking, Duration-Analyse, Heatmap
<code>load_sensitivity_data()</code>	Lädt die Szenario-CSVs aus <code>sensitivity_analysis.py</code> (parallelisiert) und kombiniert sie zu einem DataFrame für die systematische Auswertung.	data.table mit allen Szenario-Pfaden
<code>plot_sensitivity_interaction()</code>	Interaktionsplot: ein Parameter (z.B. Durchschnittsalter) gegen eine Zielgrösse (z.B. <code>P(Unterdeckung)</code>), eingefärbt nach einer Gruppe (z.B. Duration-Modus), mit LOESS-Glättung.	ggplot-Objekt
<code>run_sensitivity_analysis()</code>	Komplett-Auswertung: lädt die Szenario-Daten (parallel), berechnet Risikometriken und erstellt sämtliche Visualisierungen in einem Durchlauf.	Liste mit Rohdaten, Risikometriken und Plots

14.1 R-Analyseskripte im Projekt

Zusätzlich zu `functions.R` stehen drei eigenständige R-Skripte für die Auswertung und Visualisierung der Simulations- und Strategieergebnisse bereit.

Skript / Funktion	Zweck	Outputs
<code>cashflow_fan_chart()</code> (<code>cashflow_fan_chart.R</code>)	Einleitungs-Chart für die Sensitivitätsanalyse: Entwicklung des absoluten Renten-Cashflows (Median sowie P5/P95- und P25/P75-Band) über den Simulationshorizont aus einer einzelnen Simulation. Liest Populations-Metadaten (Bestandsgrösse, Alter, Rente, Verheiratet, Frauenanteil, Ehegattenrente) für den Untertitel.	ggplot Fan-Chart + Konsolen-Zusammenfassung
<code>liability_duration_chart.R</code>	Fan-Chart der Liability-Duration (Median, P5, P95) über den Horizont, verglichen mit einer linearen Referenzlinie (Startduration minus 1 pro Jahr). Blendet eine Populations-	ggplot Fan-Chart (PNG) + Statistiktabelle

	Infobox ein und gibt eine Jahres-Statistiktable aus.	
strategy_analysis.R	Speichereffiziente Auswertung vieler Strategie-CSVs (jede Datei einzeln geladen und sofort zu Kennzahlen verdichtet). Berechnet Performance- (Sharpe, Sortino), LDI-Surplus- (Information Ratio, Tracking Error), Drawdown- und Unterdeckungs-Kennzahlen, erstellt Rankings (geringstes Unterdeckungsrisiko, beste Information Ratio) sowie Asset-Allocation- und Heatmap-Plots.	Ranking-CSVs (strategy_summary, Top-20) + Plots

Verwendete statistische Methoden in R

- Kolmogorov-Smirnov-Test: Vergleich von Verteilungen (Default-Pfade vs. alle Pfade)
- Chi²-Test: Unabhängigkeitstest Zinsschock vs. Default
- Variationskoeffizient (CV): Streuungsmass für Cashflow-Analyse
- Pearson-Korrelation: Zusammenhang zwischen Bestandseigenschaften und CV
- LOESS-Regression: Nicht-parametrische Glättung in Scatter-Plots
- Sharpe Ratio: Risikoadjustierte Rendite (Mean Return / Std.Dev.)
- VaR/CVaR (5%): Value at Risk und Conditional VaR der Portfolio-Rendite

15. Referenz: Input-Variablen

Die Variablen des ALM-Simulators lassen sich in zwei Gruppen unterteilen: Variablen, die vom Benutzer festgelegt werden (Web-UI bzw. config_alm.py, in der Sensitivitätsanalyse systematisch variiert), und Variablen, die vom Modell aus dem Rentnerbestand berechnet werden.

15.1 Vom Benutzer festgelegte Variablen

Diese Werte werden je Simulation als Eingabe gesetzt bzw. in der Sensitivitätsanalyse über Parameter-Grids und LHS-Stichproben variiert.

Variable	Kategorie	Beschreibung
n_paths	Simulation	Anzahl der Monte-Carlo-Pfade pro Simulation.
t_horizon	Simulation	Simulationshorizont in Jahren (z.B. 40).
mu_interest_rate	Kapitalmarkt	Erwartetes Niveau des Basiszinses (Cash-Rate); $r_{1,t}$ wird jährlich unabhängig (i.i.d.) lognormal um dieses Niveau simuliert, ohne Autokorrelation des Zinsprozesses.
mu_equities	Kapitalmarkt	Erwartete jährliche Aktienrendite.
DOF_Equities	Kapitalmarkt	Freiheitsgrade für die Student-t-Verteilung (Aktien)
mu_real_estate	Kapitalmarkt	Erwartete jährliche Immobilienrendite.
DOF_RealEstate	Kapitalmarkt	Freiheitsgrade für die Student-t-Verteilung (Immobilien)
sigma_interest_rate	Kapitalmarkt	Volatilität des Basiszinses.
sigma_equities	Kapitalmarkt	Volatilität Aktien (z.B. 15%).
sigma_real_estate	Kapitalmarkt	Volatilität Immobilien.
yield_curve_slope	Zinsstruktur	Steigung der Zinsstrukturkurve (Slope) je Durationseinheit.
basis_rate_floor	Zinsstruktur	Untergrenze (Floor) für den Basiszins $r_{1,t}$.
weight_gov_bonds	Allokation	Portfolio-Gewicht Staatsanleihen.
weight_corp_bonds	Allokation	Portfolio-Gewicht Unternehmensanleihen.
weight_equities	Allokation	Portfolio-Gewicht Aktien (Art. 55 lit. b BVV 2: max 50%).
weight_real_estate	Allokation	Portfolio-Gewicht Immobilien (Art. 55 lit. c BVV 2: max 30%).
weight_alternatives	Allokation	Portfolio-Gewicht Alternative/Infrastructure Debt (Infrastrukturanlagen: max. 10% nach Art. 53 Abs. 1 lit. dbis i.V.m. Art. 55 lit. f BVV 2; alternative Anlagen: max. 15% nach Art. 55 lit. e BVV 2; Erweiterung nach Art. 50 Abs. 4 BVV 2 möglich).
technical_rate_duration	Verbindlichkeiten	Duration zur Herleitung des technischen Zinssatzes.
liability_discount_spread	Verbindlichkeiten	Liability-Discount-Spread auf den technischen Zinssatz.
technical_rate_floor	Verbindlichkeiten	Mindest-Niveau (Floor) des technischen Zinssatzes.
initial_gov_bond_duration	Bond Gov	Initiale Duration der Staatsanleihen.
gov_bond_duration_mode	Bond Gov	Duration-Modus (fixed, fixed_reset, liability_matching, cashflow_matching).
gov_bond_duration_reset_interval	Bond Gov	Reset-Intervall im Modus fixed_reset (Jahre).
initial_corp_bond_duration	Bond Corp	Initiale Duration der Unternehmensanleihen.
corp_bond_duration_mode	Bond Corp	Duration-Modus der Unternehmensanleihen.
corp_bond_duration_reset_interval	Bond Corp	Reset-Intervall im Modus fixed_reset (Jahre).
corp_bond_credit_spread	Bond Corp	Credit Spread auf Unternehmensanleihen (s_c).
corp_bond_default_probability	Bond Corp	Jährliche Ausfallwahrscheinlichkeit $p_{Default}$ (z.B. 0.3%).
corp_bond_loss_given_default	Bond Corp	Verlustquote bei Ausfall (LGD, z.B. 40%).
corp_bond_default_exposure	Bond Corp	Bei Default betroffener Portfolioanteil (Exposure).
initial_alt_bond_duration	Bond Alt	Initiale Duration der Infrastructure Debt.
alt_bond_duration_mode	Bond Alt	Duration-Modus der Infrastructure Debt.
alt_bond_duration_reset_interval	Bond Alt	Reset-Intervall im Modus fixed_reset (Jahre).
alt_bond_credit_spread	Bond Alt	Spread auf Infrastructure Debt (s_a).
alt_bond_default_probability	Bond Alt	Jährliche Ausfallwahrscheinlichkeit (z.B. 1.3%).
alt_bond_loss_given_default	Bond Alt	Verlustquote bei Ausfall (LGD).
alt_bond_default_exposure	Bond Alt	Bei Default betroffener Portfolioanteil.
general_reserve_rate	Reserve/Kosten	Wertschwankungsreserve (GRR); bestimmt Start-Deckungsgrad = 1 + GRR.
admin_fee_per_person	Reserve/Kosten	Verwaltungskosten pro Person und Jahr.
equity_mean_reversion_enabled	Mean Reversion	Aktien-Mean-Reversion aktiviert (True/False).
equity_long_term_annual_return	Mean Reversion	Langfristige jährliche Aktienrendite (μ_{LT}) als Referenz.
equity_mean_reversion_strength	Mean Reversion	Stärke der Mean-Reversion-Anpassung.
equity_mean_reversion_threshold	Mean Reversion	Shortfall-Schwelle, ab der Mean Reversion greift.
interest_rate_cap_enabled	Interest Rate Cap	Zins-Cap aktiviert (True/False).
interest_rate_cap_strike	Interest Rate Cap	Strike-Level des Caps (Zinsschwelle).
interest_rate_cap_premium	Interest Rate Cap	Prämiensatz des Caps (auf Notional).
interest_rate_cap_duration	Interest Rate Cap	Laufzeit des Caps in Jahren.
special_pension_enabled	Sonder-Rente	Sonder-Rente bei Überdeckung aktiviert (True/False).
special_pension_threshold	Sonder-Rente	Deckungsgrad-Schwelle für Auszahlung (DG_Threshold).
special_pension_target	Sonder-Rente	Ziel-Deckungsgrad nach Auszahlung (DG_Target).
sammelstiftung_enabled	Sammelstiftung	Sammelstiftungs-Modus aktiviert (True/False).

sammelstiftung_interval	Sammelstiftung	Intervall (Jahre) für Aufnahme eines neuen Bestands.
-------------------------	----------------	--

15.2 Vom Modell berechnete Variablen (Bestands-Kennzahlen)

Die bestand_*-Spalten sind keine direkten Eingaben, sondern werden von der Funktion compute_initial_population_stats() aus dem generierten bzw. geladenen Rentnerbestand abgeleitet (W(0) zusätzlich aus dem technischen Zinssatz). In der Sensitivitätsanalyse werden sie indirekt über die Populations-Parameter (z.B. n_population, age_mean, pension_mean, share_married) gesteuert.

Variable	Beschreibung
bestand_n_total	Anzahl Rentner im Anfangsbestand (= Länge des Bestands-DataFrames).
bestand_avg_age	Durchschnittsalter des Bestands (Mittelwert Age).
bestand_n_m	Anzahl Männer im Bestand.
bestand_n_f	Anzahl Frauen im Bestand.
bestand_share_m	Männeranteil (n_m / n_{total}).
bestand_share_f	Frauenanteil (n_f / n_{total}).
bestand_n_married	Anzahl Verheiratete (MaritalStatus == 'Married').
bestand_share_married	Anteil Verheiratete ($n_{married} / n_{total}$).
bestand_share_single	Anteil Alleinstehende ($1 - share_{married}$).
bestand_total_initial_pension	Summe der Anfangsrenten (CHF/Jahr).
bestand_pension_min	Minimale Jahresrente im Bestand.
bestand_pension_p10	10%-Perzentil der Jahresrenten.
bestand_pension_p25	25%-Perzentil der Jahresrenten.
bestand_pension_median	Median der Jahresrenten.
bestand_pension_mean	Durchschnittliche Jahresrente.
bestand_pension_std	Standardabweichung der Jahresrenten.
bestand_pension_p75	75%-Perzentil der Jahresrenten.
bestand_pension_p90	90%-Perzentil der Jahresrenten.
bestand_pension_max	Maximale Jahresrente im Bestand.
bestand_w0	Initialer Barwert der Verpflichtungen W(0), berechnet aus Bestand und technischem Zinssatz.
bestand_spouse_age_diff_mean	Mittlere Altersdifferenz zum Ehepartner (Mittelwert SpouseAgeDiff).
bestand_spouse_age_diff_std	Standardabweichung der Altersdifferenz zum Ehepartner.
bestand_spouse_pension_rate	Mittlerer Ehegattenrenten-Satz (α_{spouse}) im Bestand.

16. Referenz: Ergebnis-Spalten (mc_all_paths*.csv)

Die Ergebnis-CSV (mc_all_paths*.csv, Trennzeichen ';', Dezimalpunkt '.') enthält eine Zeile pro Pfad und Simulationsjahr. Die ersten Spalten sind die eigentlichen Zeitreihen-Ergebnisse:

Spalte	Beschreibung
path_nr	Laufende Pfad-Nummer (Monte-Carlo-Pfad).
year	Simulationsjahr (1 ... t_horizon).
deckungsgrad	Deckungsgrad $DG(t) = V(t) / W(t)$.
V_t	Vermögen (Assets) am Ende von Jahr t.
W_t	Barwert der Verpflichtungen (Liabilities) in Jahr t.
portfolio_return	Gewichtete Gesamtrendite des Portfolios in Jahr t.
gov_bonds_return	Gesamtrendite Staatsanleihen.
corp_bonds_return	Gesamtrendite Unternehmensanleihen.
equities_return	Aktienrendite.
realestate_return	Immobilienrendite.
alternatives_return	Rendite Infrastructure Debt.
gov_bonds_coupon	Coupon-Komponente Staatsanleihen (C_{fix} / MV).
gov_bonds_duration_effect	Kursänderung Staatsanleihen aus Zinsänderung (Duration + Konvexität).
corp_bonds_coupon	Coupon-Komponente Unternehmensanleihen.
corp_bonds_duration_effect	Duration-/Konvexitäts-Effekt Unternehmensanleihen.
corp_bonds_default_loss	Default-Verlust Unternehmensanleihen ($Exposure \cdot LGD$).
alt_bonds_coupon	Coupon-Komponente Infrastructure Debt.
alt_bonds_duration_effect	Duration-/Konvexitäts-Effekt Infrastructure Debt.
alt_bonds_default_loss	Default-Verlust Infrastructure Debt.
r1_t	Basiszins (Cash-Rate) nach Floor in Jahr t.
i_gov_bonds_t	Marktzins Staatsanleihen $i_{Gov}(t)$.
i_corp_bonds_t	Marktzins Unternehmensanleihen $i_{Corp}(t)$.
i_alt_bonds_t	Marktzins Infrastructure Debt $i_{Alt}(t)$.
i_tech_t	Technischer Zinssatz $i_{tech}(t)$ zur Diskontierung der Liabilities.
cashflow_rent	Rentenzahlungen in Jahr t (CHF).
cashflow_admin_fee	Verwaltungskosten in Jahr t (CHF).
cashflow_total	Gesamte Cashflows = Renten + Verwaltungskosten (+ Sonder-Rente).
special_pension_payout	Ausgeschüttete Sonder-Rente in Jahr t (bei Überdeckung).
liability_duration	Macaulay-Duration der Verpflichtungen $D_{Liab}(t)$.
num_pensioners	Anzahl lebender Rentner in Jahr t.
num_widows	Anzahl Witwen/Witwer (Hinterbliebenenrenten) in Jahr t.
interest_rate_cap_payout	Auszahlung aus dem Interest Rate Cap in Jahr t.
interest_rate_cap_active	1 wenn der Cap aktiv und im Geld ist, sonst 0.

Die übrigen Spalten (Index 34–112) wiederholen die konstanten Eingangs- und Bestandsparameter pro Zeile zur Nachvollziehbarkeit. Sie entsprechen den in der Input-Tabelle beschriebenen Variablen: die Asset-Gewichte (weight_*), Kapitalmarkt-Parameter (mu_*, sigma_*), Zinsstruktur (yield_curve_slope, basis_rate_floor), Liability-Parameter (technical_rate_duration, liability_discount_spread, technical_rate_floor), die Bond-Parameter je Klasse (initial_*_bond_duration, *_duration_mode, *_duration_reset_interval, *_credit_spread, *_default_probability, *_loss_given_default, *_default_exposure), Reserve/Kosten (general_reserve_rate, admin_fee_per_person), Mean-Reversion-, Interest-Rate-Cap-, Sonder-Renten- und Sammelstiftungs-Parameter, die Lauf-Parameter (n_paths, t_horizon) sowie sämtliche Bestands-Kennzahlen (bestand_*). Hinweis: Die exportierten Parameter mu_gov_bonds, mu_corp_bonds und mu_alternatives sowie die zugehörigen sigma_-Werte sind ohne Modellwirkung, da die Renditen der Anleihenklassen vollständig aus dem Bond-Modell (Kap. 4) abgeleitet werden.