# General Idea

The three phases:

1. Saving Phase

Basic idea is just that with a starting amount and a monthly invest, for a given set of different assets, the resulting amount is calculated including interests.

1. StopWorkPhase

Input is the result of Phase 1 and 3, so this phase is calculated at last.

The StopWorkPhase cannot fit between SavingPhase and RentPhase with exact the same (or inflation-adapted) rates. That’s why the condition is that the rates for this phase should be \*at least\* the minimum that is expected for this phase. It its more, then even better.

Also in this phase, we have to calculate with good case / bad case.

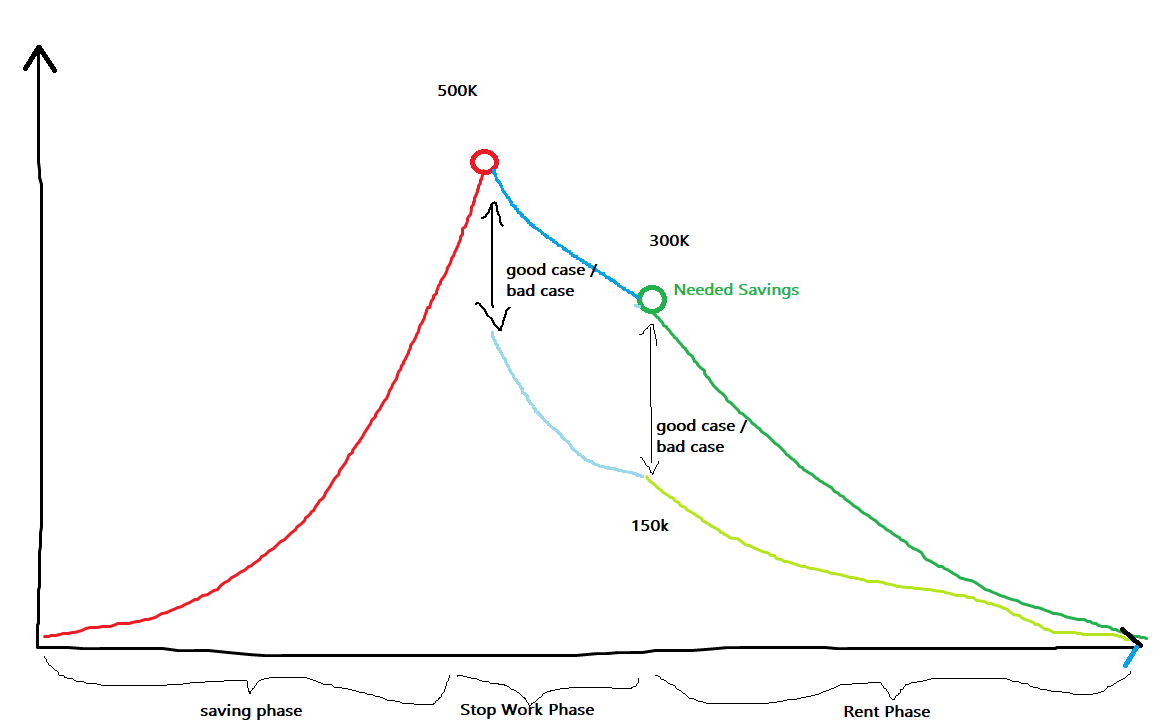
1. RentPhase

In the rent phase we may have only two remaining asset classes. One low risk asset class (typically cash), where we don’t expect crashes, but also without interests, and a high risk asset class (typically stocks) which is volatile but produces interests.

Another input is the “needs”. I know already which minimum monthly amount is required to “survive” and which monthly amount is required to live “comfortable”.

Idea is to calculate the RentPhase based on the question “What amount of low and high class asset is needed to get the comfort amount in case the high risk asset behaves normal, and to get the minimum amount in case the high risk asset behaves bad – meaning stock market crashes and interests go down.

Interpretation of the picture: The upper band is the amount which can be used with stable stocks, the lower band with a crash scenario. But this is not fixed, these are just two “limits” and the real amount to be withdrawn is somewhere inbetween and depends on the stockmarket situation. That’s why the good case / bad case consideration must be started already with the “stop work phase”.



# Dependencies

Saving Phase

StopWork Phase

Rent Phase

**A**

**B**

1. Rent Phase has a dependency to Stop Work Phase

Input: later needs / net state rent

In the rent phase, we need the net state rent (in form of later needs) as input, which depends on how long we work, means the duration of the saving phase. The longer the savong phase, the more net state rent and the higher the rent phase rate.

1. Stop Work Phase has a dependency to Rent Phase

Input: amount needed at rent start phase, rates per year

The needed amount at rent start begin is needed because it’s the target amount of the the stop work phase.

* Order of calculation: 1) Saving Phase 2) Rent Phase 3) StopWork Phase

## Glossar Zinsen

<https://www.matheboard.de/archive/2495/thread.html>

Beispiel mit 26% Zinsen

Zins**satz**: 0,26 (typisches Kürzel: i) Englisch: Interest Rate

Zins**fuß**: 26 (typisches Kürzel: p) Englisch: Interest Rate

Zins**faktor**: 1,26 (typisches Kürzel: q) Englisch: Interest Factor

Glossar:

{What}\_{AssetClass}\_{Scenario}\_{TimeRange}

What := {SavingsNeeded, WithdrawalRate, }

AssetClass := {Stocks. Cash, Total}

Scenario := { GoodCase, BadCase }

TimeRange := { PerMonth, PerYear, RentPhase}

SavingsNeeded\_Cash\_

SavingsNeeded\_Stocks\_

SavingsNeeded\_Total\_

Input:

Output:

SavingsNeeded = Der Vermögensbetrag der über die Rentenphase (n Jahre) gebraucht wird.

**Annahme:**

*TotalNeeded\_max* := Der Vermögensbetrag der über die Rentenphase (n Jahre) gebraucht wird um komfortabel zu leben

*TotalNeeded\_min* := Der Vermögensbetrag der über die Rentenphase (n Jahre) gebraucht minimal gebraucht wird.

# Rent Phase

# 

Welchen Beträge *Total\_Cash* und *Total\_Stocks* werden am Anfang des Rentenalters benötigt um

* Jährlich den Betrag *TotalNeeded\_max / n* zur Verfügung zu haben, unter der Annahme dass sich der Aktienmarkt normal positive entwickelt
* Jährlich den Betrag *TotalNeeded\_min / n* zur Verfügung zu haben, unter der Annahme dass der Aktienmarkt um einen gegebenen Faktor zu Beginn des Renteneintritts eintritt und sich nicht mehr erholt

Und wie hoch sind die jährlichen Entnahmeraten *Rate\_Cash, Rate\_Stocks\_Max, Rate\_Stocks\_Max*?

Anmerkung:

Die Cash Rate ist immer gleich da nur der Anteil der Aktien an den konstanten oder einbrechenden Aktienmarkt anpassen muss.

Die jährliche Rate wird mit einem Faktor multipliziert der sich aus der Sparkassenformel ergibt:

Kn = K0 \* q^n – R \* (q^n -1)/(q-1)

Kn ist 0, da von Kapitalverzehr ausgegangen wird.

K0 ist das Anfangskapital, R die jährliche Entnahmerate.

Die Formal muss nach K0 aufgelöst werden so dass

K0 = R \* z

gilt.

Z ist der Faktor der bei n Jahren Laufzeit den Zinseszins beinhaltet. Angenommen n=13 und q=1.0, d.h. das Vermögen wächst garnicht, dann wäre z=n=13, d.h. man würde einfach das Anfangskapital durch die Anzahl der Jahre teilen. Je höher der Zinssatz desto niedriger der Faktor. Bei 8% Zinsen und q=1.08 wäre z bei ca 7, d.h. man kann durch den Zinseszins eine (hohe) Rate von K0/7 statt K0/13 pro Jahr abheben.

Z = -(p^n - 1) / ((p-1) \* (p^n)));

Berechnen mit sympy (<https://live.sympy.org/>)

Kn, K0, q, n = symbols('Kn K0 q n')

eq1 = Eq(K0 \* q\*\*n - (K0/n)\*(q\*\*n-1)/(q-1), Kn)

sol = solve(eq1,K0)

Damit kann man nun die Zinsfaktoren

z\_stocks\_max

z\_stocks\_min

z\_cash

vorberechnen.

K0 = R \* -(q\*\*n-1) / ( (q-1) \* (q\*\*n) )

K0 = R\_cash \* z\_cash

K0 = R\_stocks \* z\_stocks\_max

K0 = R\_cash \* z\_cash

K0 = (R\_stocks \* z\_stocks\_min) / CrashFactor

z\_stocks\_max= -( q\_stocks\_max \*\*n-1) / (( q\_stocks\_max -1)\*( q\_stocks\_max \*\*n))

z\_stocks\_min= -( q\_stocks\_min \*\*n-1) / (( q\_stocks\_min -1)\*( q\_stocks\_min \*\*n))

z\_cash = -( q\_cash \*\*n-1) / (( q\_cash -1)\*( q\_cash \*\*n))

Für die finalen Raten kann man nun ein Gleichungssystem mit 5 Gleichung und 5 Unbekannten aufstellen:

1. **Total\_Stocks = Rate\_Stocks\_Max** \* z\_stocks\_max
2. **Total\_Stocks** \* stocks\_crashFactor\_badCase = **Rate\_Stocks\_Min** \* z\_stocks\_min
3. **Total\_Cash** **= Rate\_Cash** \* z\_cash
4. **Rate\_Cash** + **Rate\_Stocks\_Max** = comfort\_total\_needed\_Year
5. **Rate\_Cash** + **Rate\_Stocks\_Min** = minimum\_total\_needed\_Year

Mit sympy:

Rate\_Cash, z\_cash, Rate\_Stocks\_Max, Rate\_Stocks\_Min, z\_stocks\_max, z\_stocks\_min, Total\_Cash, Total\_Stocks, stocks\_crashFactor\_badCase, comfort\_total\_needed\_Year, minimum\_total\_needed\_Year = symbols('Rate\_Cash z\_cash Rate\_Stocks\_Max Rate\_Stocks\_Min z\_stocks\_max z\_stocks\_min Total\_Cash Total\_Stocks stocks\_crashFactor\_badCase comfort\_total\_needed\_Year minimum\_total\_needed\_Year ')

//eq1=Eq(Rate\_Cash \* z\_cash + Rate\_Stocks\_Max \* z\_stocks\_max, Total\_Cash + Total\_Stocks)

//eq2=Eq(Rate\_Cash \* z\_cash + Rate\_Stocks\_Min \* z\_stocks\_min, Total\_Cash + (Total\_Stocks\* stocks\_crashFactor\_badCase))

eq1=Eq(Rate\_Stocks\_Max \* z\_stocks\_max, Total\_Stocks)

eq2=Eq(Rate\_Stocks\_Min \* z\_stocks\_min, Total\_Stocks\* stocks\_crashFactor\_badCase)

eq3=Eq(Total\_Cash, Rate\_Cash \* z\_cash)

eq4=Eq(Rate\_Cash + Rate\_Stocks\_Max, comfort\_total\_needed\_Year)

eq5=Eq(Rate\_Cash + Rate\_Stocks\_Min, minimum\_total\_needed\_Year)

sol = solve((eq1, eq2, eq3, eq4, eq5),( Rate\_Cash, Rate\_Stocks\_Min, Rate\_Stocks\_Max, Total\_Cash, Total\_Stocks))

sol

Ergebnis:

Rate\_Cash: (comfort\_total\_needed\_Year\*stocks\_crashFactor\_badCase\*z\_stocks\_max - minimum\_total\_needed\_Year\*z\_stocks\_min)/(stocks\_crashFactor\_badCase\*z\_stocks\_max - z\_stocks\_min),

Rate\_Stocks\_Max: z\_stocks\_min\*(-comfort\_total\_needed\_Year + minimum\_total\_needed\_Year)/(stocks\_crashFactor\_badCase\*z\_stocks\_max - z\_stocks\_min),

Rate\_Stocks\_Min: stocks\_crashFactor\_badCase\*z\_stocks\_max\*(-comfort\_total\_needed\_Year + minimum\_total\_needed\_Year)/(stocks\_crashFactor\_badCase\*z\_stocks\_max - z\_stocks\_min),

Total\_Cash: z\_cash\*(comfort\_total\_needed\_Year\*stocks\_crashFactor\_badCase\*z\_stocks\_max - minimum\_total\_needed\_Year\*z\_stocks\_min)/(stocks\_crashFactor\_badCase\*z\_stocks\_max - z\_stocks\_min),

Total\_Stocks: z\_stocks\_max\*z\_stocks\_min\*(-comfort\_total\_needed\_Year + minimum\_total\_needed\_Year)/(stocks\_crashFactor\_badCase\*z\_stocks\_max - z\_stocks\_min)}

# Arbeitshypothese: Steuern mitberechnen

Herleitung laut <https://de.wikipedia.org/wiki/Sparkassenformel#:~:text=Als%20Sparkassenformeln%20werden%20in%20der,(jeweils%20pro%20Periode)%20herstellen>.

S= SteuerFaktor, e.g. 1.26

Neue, etwas angepasste Sparkassenformel:

Wegen K\_n = 0 bleibt

Und jetze einfach R mit Steuersatz multiplizieren und S=1.26

Für sympy:

K0, R, S, q, n = symbols('K0 R S q n')

eq1 = (0, K0 \* q\*\*n - R \* S \* (q\*\*n - 1) / (q - 1))

sol = solve(eq1, R)

sol

ergebnis:

{R: K0\*q\*\*n \*(q - 1)/(S\*(q\*\*n - 1))}

Passt! Simulation war erfolgreich!

Jetzt nochmal wie oben:

{K0: R\*S\*q\*\*(-n)\*(q\*\*n - 1)/(q - 1)}

Die 3 Fälle, mit substitution:

mit

mit

mit

Für die finalen Raten kann man nun ein Gleichungssystem mit 5 Gleichung und 5 Unbekannten aufstellen:

1. **Rate\_Stocks\_Max** \* z\_stocks\_max = **Total\_Stocks**
2. **Rate\_Stocks\_Min** \* z\_stocks\_min **= Total\_Stocks** \* stocks\_crashFactor\_badCase
3. **Rate\_Cash** \* z\_cash = **Total\_Cash**
4. **Rate\_Cash** + **Rate\_Stocks\_Max** = comfort\_total\_needed\_Year
5. **Rate\_Cash** + **Rate\_Stocks\_Min** = minimum\_total\_needed\_Year

# Stop Work Phase

Sparkassenformel:

Neue, etwas angepasste Sparkassenformel, mit S= SteuerFaktor, e.g. 1.26

In dieser Phase sind K\_0 und K\_n ja bekannt, K\_0 ist das was aus der Saving Phase rauskommt, K\_n ist das Anfangskapital was ich in der Rent phase brauche.

Auch die monatliche Rate R ist bekannt

Gesucht ist n.

Mal schauen was bei einer Umformung rauskommt:

Sympy:

K0, Kn, R, S, q, n = symbols('K0 Kn R S q n')

Eq1=Eq(Kn, K0\*q\*\*n - R\*S\*(q\*\*n-1)/(q-1))

sol = solve(Eq1, n)

sol

Results:

[(-K0\*q\*\*n + K0\*q\*\*(n + 1) - Kn\*q + Kn)/(S\*(q\*\*n - 1))]

Nach n aufgelöst:

[log((-Kn\*q + Kn + R\*S)/(-K0\*q + K0 + R\*S))/log(q)]

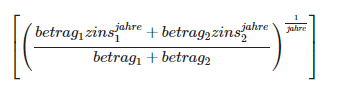
# Kombinierter Zins

betrag1, betrag2, zins1, zins2, jahre, x = symbols('betrag1 betrag2 zins1 zins2 jahre x')

eq1 = Eq(betrag1 \* zins1\*\*jahre + betrag2 \* zins2\*\*jahre, (betrag1 + betrag2) \* x\*\*jahre)

sol = solve(eq1, x)

sol



# Calculation of stocks taxes

(rateGesamt + tax) \* factorDyn = rateStocksFromI + rateStocksFromD + tax

rateStocksFromI + rateStocksFromD = rateGesamt - rateCash - rateMetals

tax = (rateStocksFromI - TAXFREEAMOUNTPERYEAR) \* CAPITALYIELDSTAXFACTOR

rateGesamt, tax, factorDyn, rateStocksFromI, rateStocksFromD, rateCash, rateMetals, TAXFREEAMOUNTPERYEAR, CAPITALYIELDSTAXFACTOR = symbols('rateGesamt tax factorDyn rateStocksFromI rateStocksFromD rateCash rateMetals TAXFREEAMOUNTPERYEAR CAPITALYIELDSTAXFACTOR')

eq1 = Eq((rateGesamt + tax) \* factorDyn, rateStocksFromI + rateStocksFromD + tax)

eq2 = Eq( rateStocksFromI + rateStocksFromD, rateGesamt - rateCash - rateMetals)

eq3 = Eq(tax, (rateStocksFromI - TAXFREEAMOUNTPERYEAR) \* CAPITALYIELDSTAXFACTOR)

sol = solve ((eq1, eq2, eq3), (tax, rateStocksFromD, rateStocksFromI))

str(sol)

Result:

tax:

(-factorDyn\*rateGesamt - rateCash + rateGesamt - rateMetals)/(factorDyn - 1),

rateStocksFromD: (-CAPITALYIELDSTAXFACTOR\*TAXFREEAMOUNTPERYEAR\*factorDyn + CAPITALYIELDSTAXFACTOR\*TAXFREEAMOUNTPERYEAR - CAPITALYIELDSTAXFACTOR\*factorDyn\*rateCash + CAPITALYIELDSTAXFACTOR\*factorDyn\*rateGesamt - CAPITALYIELDSTAXFACTOR\*factorDyn\*rateMetals + CAPITALYIELDSTAXFACTOR\*rateCash - CAPITALYIELDSTAXFACTOR\*rateGesamt + CAPITALYIELDSTAXFACTOR\*rateMetals + factorDyn\*rateGesamt + rateCash - rateGesamt + rateMetals)/(CAPITALYIELDSTAXFACTOR\*factorDyn - CAPITALYIELDSTAXFACTOR),

rateStocksFromI: (CAPITALYIELDSTAXFACTOR\*TAXFREEAMOUNTPERYEAR\*factorDyn - CAPITALYIELDSTAXFACTOR\*TAXFREEAMOUNTPERYEAR - factorDyn\*rateGesamt - rateCash + rateGesamt - rateMetals)/(CAPITALYIELDSTAXFACTOR\*factorDyn - CAPITALYIELDSTAXFACTOR)}'

My Portfolio consists of 3 asset classes: 20% cash, 50% stocks and 30% precious metals.

I want to withdraw a yearly fix rate X of my portfolio, maintaining the rate to consist of 20% cash, 50% stocks and 30% precious metals. However, on withdrawal of stocks, I have to pay 26% taxes.

How can I calculate the amount that I have to withdraw, which already contains the taxes, such that the assets without taxes sum up to X?

rate\_stocks + taxes\_stocks = factorStocksDyn \* input.TotalRateNeeded\_PerYear

taxes\_stocks = (rate\_stocks-1000) \* 0.03

rate\_stocks + pos(rate\_stocks-1000)\*0.03 = factorStocksDyn \* input.TotalRateNeeded\_PerYear

a + abs(a-tf)\*z = b\*c

a, b, c, tf, z = symbols('a b c tf z', real=True)

eq1 = Eq(a + abs(a - tf)\*z, b\*c)

sol = solve(eq1, a)

sol

result:

(-bc+tfz)/(z-1)

oder

(bc+tfz)/(z+1)

rate\_stocks = (-(factorStocksDyn \* input.TotalRateNeeded\_PerYear) + 1000\*0.03) / (0.03 - 1)

rate\_stocks = ((factorStocksDyn \* input.TotalRateNeeded\_PerYear) + 1000\*0.03) / (0.03 + 1)