



This work is licensed under a [Creative Commons](https://creativecommons.org/licenses/by-nc-sa/4.0/)

[NonCommercial Sampling Plus 1.0 License](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Finanzmathematik-Paket



Dokumentnummer: D2036
Fachgebiet: Finanzmathematik
Einsatz: zweites Lernjahr (3HAK)

Wenn man in der Maxima-Hilfe stöbert, findet man auch ein Finanzmathematikpaket. Einige Funktionen sind gut verwendbar!

```
(%i1) kill(all)$
```

```
(%i1) load(finance)  
/* Laden des Unterprogramms */$
```

Funktion:
days360 (Jahr1, Monat1, Tag1, Jahr2, Monat2, Tag2)
Jahr : 360 Tage
Monat: 30 Tage

```
(%i2) days360(2012,1,1,2012,12,31);
```

```
(%o2) 360
```

Funktion:
fv (i, AK, n)
fv = Endkapital
(diese Funktion hat keinen didaktischen Vorteil)

```
(%i3) fv(0.03,1000,10);
```

```
(%o3) 1343.916379344122
```

```
(%i4) Kn:1000*1.03**10;
```

```
(%o4) 1343.916379344122
```

Funktion:
pv (i, EK, n)
pv = Anfangskapital
(diese Funktion hat keinen didaktischen Vorteil)

```
(%i5) pv(0.03,1343.92,10);
```

```
(%o5) 1000.002694108007
```

```
(%i6) K0:1343.92/1.03**10;
```

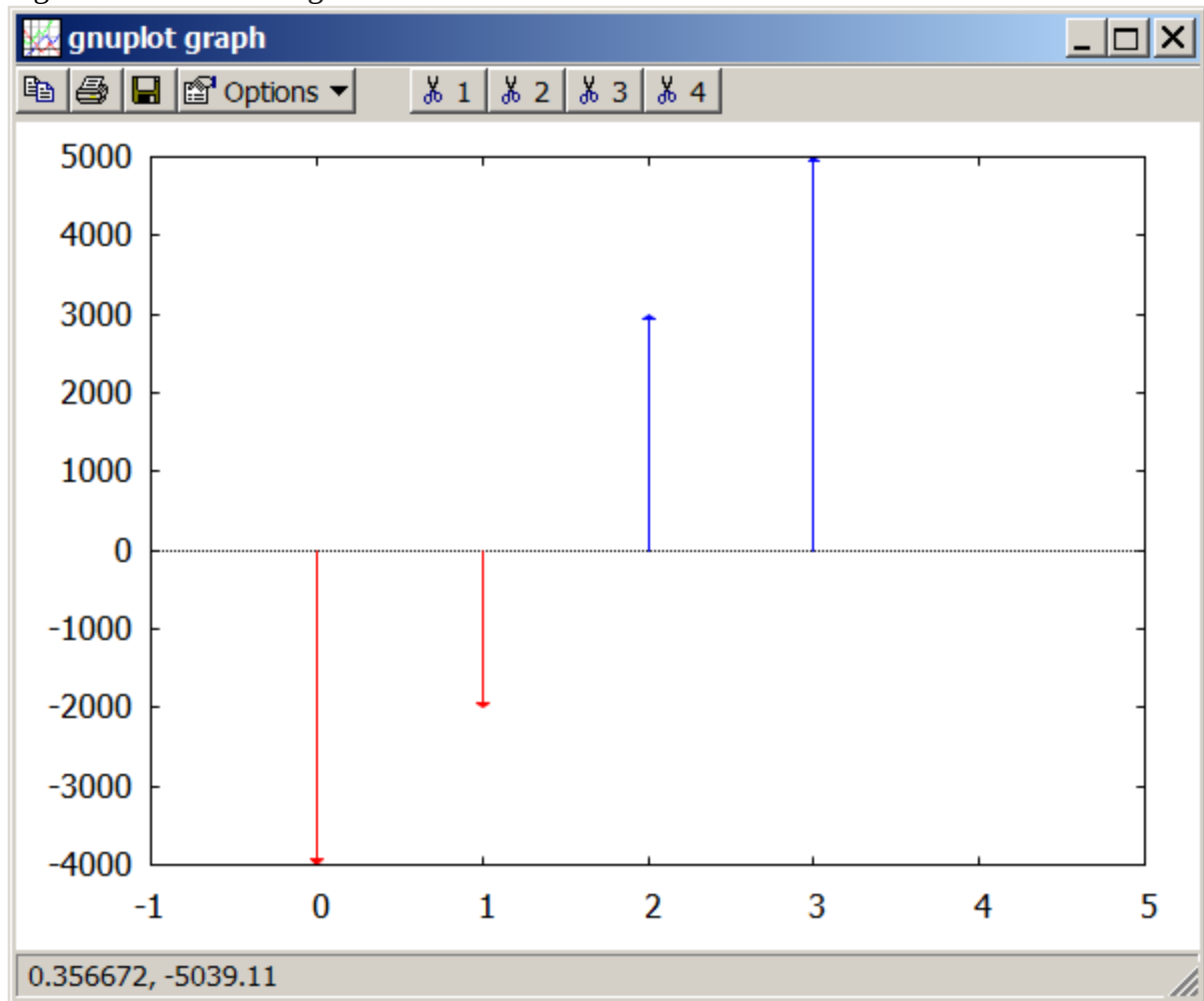
```
(%o6) 1000.002694108007
```

Funktion:
graph_flow (Werteliste)
Erzeugt eine Zeitlinie der Geldflüsse

```
(%i7) graph_flow([-4000,-2000,3000,5000]);
```

```
(%o7) [gr2d(vector, vector, vector, vector)]
```

Figure 1: Das ist das Ergebnis



Funktionen:

`annuity_pv (i, BW, n)`

$BW = R \cdot (1 - v^n) / i \rightarrow R = BW \cdot i / (1 - v^n)$

Berechnung der Raten aus dem Barwert

`annuity_fv (i, EW, n)`

$EW = R \cdot (r^n - 1) / i \rightarrow R = EW \cdot i / (r^n - 1)$

Berechnung der Raten aus dem Endwert

```
(%i8) annuity_pv(0.08,100000,20);
```

```
(%o8) 10185.22088231506
```

```
(%i9) R:100000*0.08/(1-1/1.08**20);
```

```
(%o9) 10185.22088231506
```

```
(%i10) annuity_fv(0.08,100000,20);
```

```
(%o10) 2185.220882315059
```

```
(%i11) R:100000*0.08/(1.08**20-1);
```

```
(%o11) 2185.220882315059
```

Funktionen:

geo_annuity_pv (i, j, BW, n)

geo_annuity_fv (i, j, EW, n)

Die regelmäßigen Zahlungen unterliegen einem bestimmten Wachstumsfaktor!

```
(%i12) geo_annuity_pv(0.05,0.03,100000,10);
```

```
(%o12) 11431.71200923731
```

Figure 2: Kontrollrechnung

	A	B	C
1	1	11.431,71	17.734,33
2	2	11.774,66	17.396,54
3	3	12.127,90	17.065,17
4	4	12.491,74	16.740,12
5	5	12.866,49	16.421,26
6	6	13.252,49	16.108,48
7	7	13.650,06	15.801,65
8	8	14.059,56	15.500,67
9	9	14.481,35	15.205,42
10	10	14.915,79	14.915,79
11			162.889,43
12			99.999,98

```
(%i13) geo_annuity_fv(0.05,0.03,100000,10);
```

```
(%o13) 7018.079513131849
```

Figure 3: Kontrollrechnung

	A	B	C
1	1	7.018,08	10.887,35
2	2	7.228,62	10.679,97
3	3	7.445,48	10.476,54
4	4	7.668,85	10.276,99
5	5	7.898,91	10.081,23
6	6	8.135,88	9.889,21
7	7	8.379,95	9.700,84
8	8	8.631,35	9.516,07
9	9	8.890,29	9.334,81
10	10	9.157,00	9.157,00
11			100.000,01

Funktion:
amortization (i, BW, n)
TILGUNGSPLAN

(%i14) amortization(0.08,100000,10)\$

"n"	"Balance"	"Interest"	"Amortization"	"Payment"
0.000	100000.000	0.000	0.000	0.000
1.000	93097.051	8000.000	6902.949	14902.949
2.000	85641.866	7447.764	7455.185	14902.949
3.000	77590.267	6851.349	8051.600	14902.949
4.000	68894.539	6207.221	8695.728	14902.949
5.000	59503.154	5511.563	9391.386	14902.949
6.000	49360.457	4760.252	10142.697	14902.949
7.000	38406.345	3948.837	10954.112	14902.949
8.000	26575.903	3072.508	11830.441	14902.949
9.000	13799.027	2126.072	12776.877	14902.949
10.000	0.000	1103.922	13799.027	14902.949

arit_amortization (i, Wachstumsbetrag, BW, n)
Linearer Zuwachs der regelmäßigen Zahlung

geo_amortization (i, Wachstumsrate, BW, n)
Exponentieller Zuwachs der regelmäßigen Zahlung

(%i15) arit_amortization(0.08,1000,100000,10)\$

Unterprogramm_Finanzmathematik

"n"	"Balance"	"Interest"	"Amortization"	"Payment"
0.000	100000.000	0.000	0.000	0.000
1.000	96968.365	8000.000	3031.635	11031.635
2.000	92694.199	7757.469	4274.166	12031.635
3.000	87078.100	7415.536	5616.099	13031.635
4.000	80012.713	6966.248	7065.387	14031.635
5.000	71382.095	6401.017	8630.618	15031.635
6.000	61061.028	5710.568	10321.067	16031.635
7.000	48914.275	4884.882	12146.753	17031.635
8.000	34795.783	3913.142	14118.493	18031.635
9.000	18547.810	2783.663	16247.972	19031.635
10.000	0.000	1483.825	18547.810	20031.635

(%i16) `geo_amortization(0.08,0.03,100000,10)$`

"n"	"Balance"	"Interest"	"Amortization"	"Payment"
0.000	100000.000	0.000	0.000	0.000
1.000	94755.201	8000.000	5244.799	13244.799
2.000	88693.475	7580.416	6061.726	13642.143
3.000	81737.546	7095.478	6955.929	14051.407
4.000	73803.601	6539.004	7933.945	14472.949
5.000	64800.751	5904.288	9002.849	14907.137
6.000	54630.460	5184.060	10170.291	15354.352
7.000	43185.915	4370.437	11444.545	15814.982
8.000	30351.356	3454.873	12834.558	16289.432
9.000	16001.350	2428.108	14350.006	16778.115
10.000	0.000	1280.108	16001.350	17281.458

Funktion:
`saving(i, EW, n)`

(%i17) `saving(0.08,100000,10)$`

"n"	"Balance"	"Interest"	"Payment"
0.000	0.000	0.000	0.000
1.000	6902.949	0.000	6902.949
2.000	14358.134	552.236	6902.949
3.000	22409.733	1148.651	6902.949
4.000	31105.461	1792.779	6902.949
5.000	40496.846	2488.437	6902.949
6.000	50639.543	3239.748	6902.949
7.000	61593.655	4051.163	6902.949
8.000	73424.097	4927.492	6902.949
9.000	86200.973	5873.928	6902.949
10.000	100000.000	6896.078	6902.949

Funktion:
npv (i, Zahlungsliste)
Zahlungen sind nachschüssig
(diese Funktion hat keinen didaktischen Vorteil)

```
(%i18) npv(0.05,[1000,2000,3000,4000,5000]),numer;
```

```
(%o18) 12566.3934364013
```

```
(%i19) 1000/1.05+2000/1.05**2+3000/1.05**3+4000/1.05**4+5000/1.05**5;
```

```
(%o19) 12566.3934364013
```

Funktion:
irr (Zahlungsliste, IO)
(irr = interner Zinsfuß,
didaktisch ist der Ansatz einer Gleichung nach dem Äquivalenzprinzip
vorzuziehen!)

```
(%i20) irr([-1000,100,120,90,110,1400],0);
```

```
(%o20) [ [ r=0.14885001619695 ] ]
```

```
(%i21) g:1000*r**5=100*r**4+120*r**3+90*r**2+110*r+1400;
```

$$(\%o21) \quad 1000 r^5 = 100 r^4 + 120 r^3 + 90 r^2 + 110 r + 1400$$

(%i22) `realroots(g),numer;`

(%o22) `[ r=1.148850053548813 ]`

Funktion:
`benefit_cost (i, input-Liste, output-Liste)`

(%i23) `benefit_cost(0.05,[200,200,300],[150,200,300]);`

(%o23) `1.081621321488062`

(%i24) `a:200/1.05+200/1.05**2+300/1.05**3;`

(%o24) `631.0333657272433`

(%i25) `b:150/1.05+200/1.05**2+300/1.05**3;`

(%o25) `583.4143181081956`

(%i26) `a/b;`

(%o26) `1.081621321488062`

Created with [wxMaxima](#).