

```

In[1]:= (* =====
          数理安全保障モデル（輸入依存度 k 版）：修正版
          Associationではなく直接変数で定義=====*)
(*▼ パラメータ設定*)
alpha0 = 10.0 * 10^6;
beta = -0.2;
r = 17;
k = 0.2;

t0 = 0; t1 = 50; cho = 10^12;
FMprod0 = 550. cho; FMwprod0 = 5500. cho;

t10 = 1.45; gT1 = 0.01;
Gprod0 = 1000. cho; gProd = 0.00;
Gtrans0 = 300. cho; gTran = 0.00;
Cprivprod0 = 4000. cho; gprivprod = 0.003;
Cprivtrans0 = 1000. cho; gprivtran = 0.0005;
Gwprod0 = 1000. cho; gwProd = 0.05;
Gwtrans0 = 500. cho; gwTran = 0.01;
pop0 = 124 000 000;
Cprivprod[t_] := Cprivprod0 Exp[gprivprod t];
|指数関数
Cprivtrans[t_] := Cprivtrans0 Exp[gprivtran t];
|指数関数
l = 0.6 (*7.7 10^7/124000000*);
(*▼ 外生系列の定義（必要に応じて差し替え）*)
t1[t_] := t10 Exp[gT1 t];
|指数関数
alpha[t_] := alpha0 t1[t]^beta;

Gprod[t_] := Gprod0 Exp[gProd t];
|指数関数
Gtrans[t_] := Gtrans0 Exp[gTran t];
|指数関数
Gwprod[t_] := Gwprod0 Exp[gwProd t];
|指数関数
Gwtrans[t_] := Gwtrans0 Exp[gwTran t];
|指数関数
dGprod[t_] := D[Gprod[t], t];
|微分
dCprivprod[t_] := Evaluate[shockCprod[t] * D[Cprivprod[t], t]];
|評価 |微分
dGtrans[t_] := D[Gtrans[t], t];
|微分
dCprivtrans[t_] := Evaluate[shockCtrans[t] * D[Cprivtrans[t], t]];
|評価 |微分
CAprod[t_] := 0;
CAtrans[t_] := 0;
dCAprod[t_] := D[CAprod[t], t];
|微分
dCAtrans[t_] := D[CAtrans[t], t];
|微分

```

```

(*金融ショックのパラメータ*)
tShockCprod = 10;
(*ショック中心時点(年)*)
ampCprod = 5; (*落ち幅: 0.3→30% 減少*)
widthCprod = 2; (*ショックの「幅」: 大きいほどゆっくり*)

tShockCtrans = 10;
ampCtrans = 5;
widthCtrans = 2;

(*ショック係数: 1 から一時的に (1-amp) まで落ちる*)
shockCprod[t_] := 1 - ampCprod * Exp[-((t - tShockCprod) ^ 2 / (2 widthCprod ^ 2))];
(*指数関数*)

shockCtrans[t_] := 1 - ampCtrans * Exp[-((t - tShockCtrans) ^ 2 / (2 widthCtrans ^ 2))];
(*指数関数*)

FMwtranssol[t_] := Evaluate[dGwtrans[t]];
(*評価*)
dGwprod[t_] := D[Gwprod[t], t];
(*微分*)
dGwtrans[t_] := D[Gwtrans[t], t];
(*微分*)

(*▼ ODE系の定義*)
Clear[FMprod, FMwprod];
(*クリア*)
FMtranssol[t_] := Evaluate[dGtrans[t] + dCprivtrans[t] + dCAtrans[t]];
(*評価*)
FMwtranssol[t_] := Evaluate[dGwtrans[t]];
(*評価*)

beta0 = 0.03;
hstar = 0.02;
kappa = 6;
delta0 = 0.010012;
lambda = 0.4;
sigma[x_] := 1 / (1 + Exp[-kappa (x - hstar)]);
(*指数関数*)

X[t_] := (dGprod[t] + dCprivprod[t] +
  FMprod[t] + FMtranssol[t] + k (FMwprod[t] + FMwtranssol[t]));
Pf[t_] := X[t] / (t1[t] × FMprod[t]);
Pl[t_] := X[t] / (alpha[t] × Lmax[t]);
EF[t_] := FMprod[t] / (alpha[t] × P[t]);
P[t_] := (Pf[t] ^ r + Pl[t] ^ r) ^ (1 / r);
Emraw[t_] := X[t] / (alpha[t] × P[t]);
RGDP[t_] := (FMprod[t] + FMtranssol[t]) / P[t];
Eprod[t_] := ((Emraw[t]) ^ (-r) + (EF[t]) ^ (-r)) ^ (-1 / r);
Etrans[t_] := Emraw[t] - Eprod[t];
hp[t_] := alpha[t] × P[t] × Emraw[t] / Popu[t] - alpha[t] × Pf[t];
hr[t_] := P[t] × Emraw[t] / (Popu[t] × Pf[t]) - 1;
RGDPobs[t_] := X[t] / P[t];

```

```

FMobs[t_] := (dGprod[t] + dCprivprod[t] +
  FMprod[t] + FMtranssol[t] + k (FMwprod[t] + FMwtranssol[t]));
sstable[t_] := Eprod[t] / Emraw[t];
Birth[t_] := beta0 Eprod[t] × sigma[hr[t]];
Death[t_] := delta0 Popu[t] Exp[-lambda hr[t]];
|指数関数

Lmax[t_] := 1 Popu[t];
eqs = {FMprod'[t] + (t1'[t] / t1[t]) FMprod[t] == dGprod[t] + dCprivprod[t] + dCAprod[t],
  FMwprod'[t] + (t1'[t] / t1[t]) FMwprod[t] == dGwprod[t], FMprod[t0] == FMprod0,
  FMwprod[t0] == FMwprod0, Popu'[t] == Birth[t] - Death[t], Popu[t0] == pop0};

sol = NDSolve[eqs, {FMprod, FMwprod, Popu}, {t, t0, t1}, MaxSteps → 10^6, AccuracyGoal → 5,
|微分方程式の数値解 |最大ステップ数 |確度目標
  PrecisionGoal → 5, WorkingPrecision → MachinePrecision, Method → Automatic];
|精度目標 |作業精度 |機械精度 |ソッド |自動

FMprodsol[t_] := Evaluate[FMprod[t] /. sol[[1]]];
|評価

FMwprodsol[t_] := Evaluate[FMwprod[t] /. sol[[1]]];
|評価

Popusol[t_] := Evaluate[Popu[t] /. sol[[1]]];
|評価

(*▼ 派生変数の定義*)
Psol[t_] := Evaluate[P[t] /. sol[[1]]];
|評価

Pfsol[t_] := Evaluate[Pf[t] /. sol[[1]]];
|評価

Plsol[t_] := Evaluate[Pl[t] /. sol[[1]]];
|評価

EFsol[t_] := Evaluate[EF[t] /. sol[[1]]];
|評価

RGDPsol[t_] := Evaluate[RGDP[t] /. sol[[1]]];
|評価

Emrawsol[t_] := Evaluate[Emraw[t] /. sol[[1]]];
|評価

Eprodsol[t_] := Evaluate[Eprod[t] /. sol[[1]]];
|評価

Eprodshadow[t_] := Evaluate[Min[Emrawsol[t], EFsol[t]]];
|評価 |最小

Etranssol[t_] := Evaluate[Etrans[t] /. sol[[1]]];
|評価

hpsol[t_] := Evaluate[hp[t] /. sol[[1]]];
|評価

hrsol[t_] := Evaluate[hr[t] /. sol[[1]]];
|評価

RGDPobssol[t_] := Evaluate[RGDPobs[t] /. sol[[1]]];
|評価

FMobssol[t_] := Evaluate[FMobs[t] /. sol[[1]]];
|評価

sstablesol[t_] := Evaluate[sstable[t] /. sol[[1]]];
|評価

```

```

Lmaxsol[t_] := Evaluate[Lmax[t] /. sol[[1]]];
評価
dCprivprodsol[t_] := Evaluate[dCprivprod[t] /. sol[[1]]];
評価
dCprivtranssol[t_] := Evaluate[dCprivtrans[t] /. sol[[1]]];
評価
(*Em[t_] := Evaluate[Min[FMsol[t] / (alpha[t] Psol[t]), Lmax[t]]];*)
評価 最小
V[t_] := (FMprodsol[t] + FMtranssol[t]) / (Gprod[t] + Gtrans[t]);

(*▼ プロット*)
plt1 = Plot[{(FMprodsol[t] + FMtranssol[t]), (FMwprodsol[t] + FMwtranssol[t])},
プロット
{t, t0, t1}, PlotLegends → {"FM (国内)", "FMw (世界)"}, PlotLabel → "名目フロー";
プロットの凡例 プロットラベル

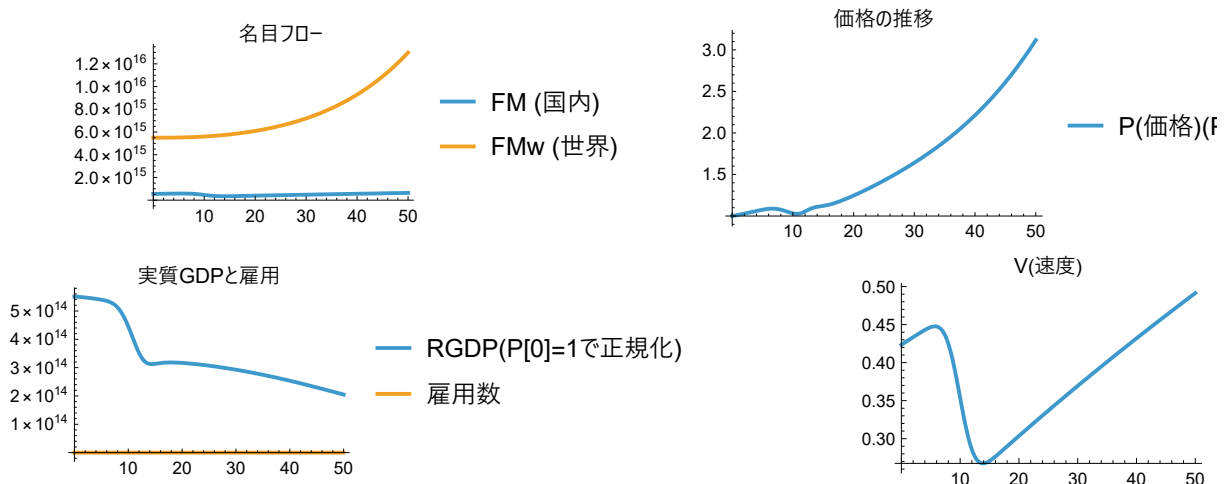
plt2 = Plot[{Psol[t] / Psol[0]}, {t, t0, t1},
プロット
PlotLegends → {"P(価格) (P[0]で正規化)"}, PlotLabel → "価格の推移";
プロットの凡例 プロットラベル

plt3 = Plot[{RGDPsol[t] × Psol[0], Emrawsol[t]}, {t, t0, t1}, PlotLegends →
プロット プロットの凡例
{"RGDP (P[0]=1で正規化)", "雇用数"}, PlotLabel → "実質GDPと雇用", PlotRange → All];
プロットラベル プロット範囲 すべて

Grid[{ {plt1, plt2}, {plt3, Plot[V[t], {t, t0, t1}, PlotLabel → "V(速度)"] } } ]
格子 プロット プロットラベル

```

Out[86]=



In[87]:= `Plot[{dCprivprodsol[t], dCprivtranssol[t]}, {t, t0, t1},`

`プロット`

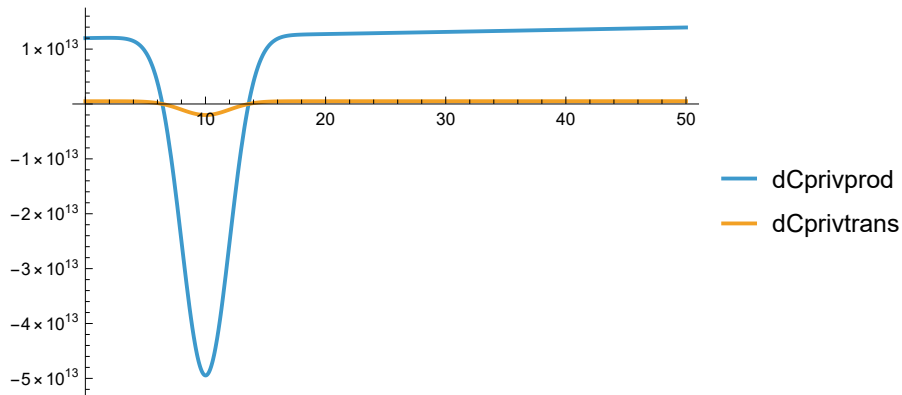
`PlotLegends → {"dCprivprod", "dCprivtrans"}, PlotRange → All]`

`プロットの凡例`

`プロット範囲`

`すべて`

Out[87]=



In[88]:= `alpha[0] × Psol[0]`

Out[88]=

2.24686×10^7

In[89]:= `k (FMwprodsol[0] + FMwtranssol[0])`

Out[89]=

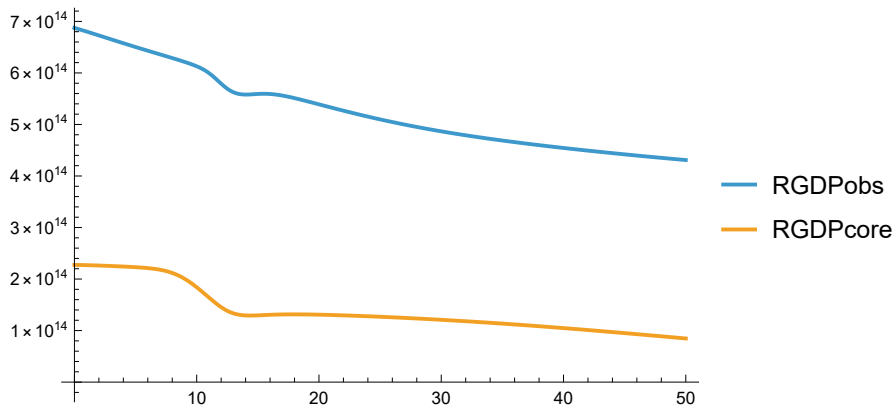
1.101×10^{15}

In[90]:= `Plot[{RGDPobssol[t], RGDPsol[t]}, {t, t0, t1}, PlotLegends → {"RGDPobs", "RGDPcore"}]`

`プロット`

`プロットの凡例`

Out[90]=



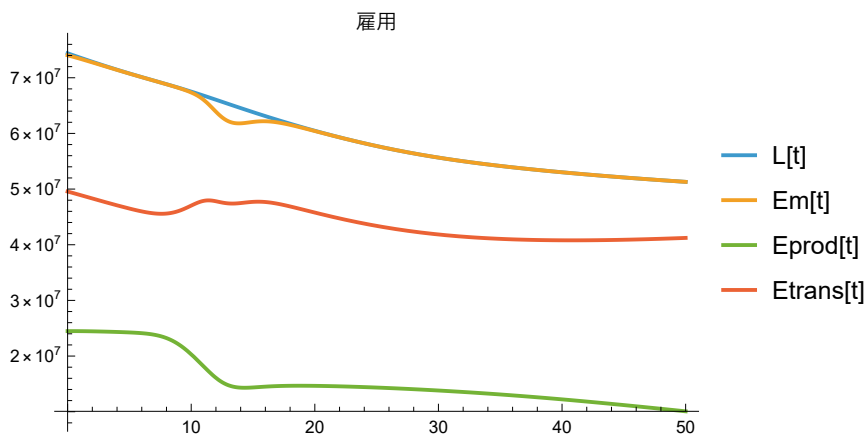
In[91]:= `Psol[0]`

Out[91]=

2.4202

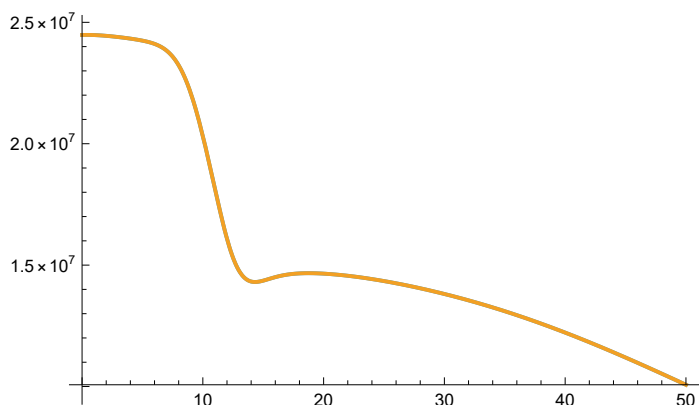
```
In[92]:= Plot[{Lmaxsol[t], Emrawsol[t], Eprodsl[t], Etranssol[t]},
プロット
  {t, t0, t1}, PlotLegends → {"L[t]", "Em[t]", "Eprod[t]", "Etrans[t]"},
プロットの凡例
  PlotLabel → "雇用", PlotRange → All]
プロットラベル プロット範囲 すべて
```

Out[92]=



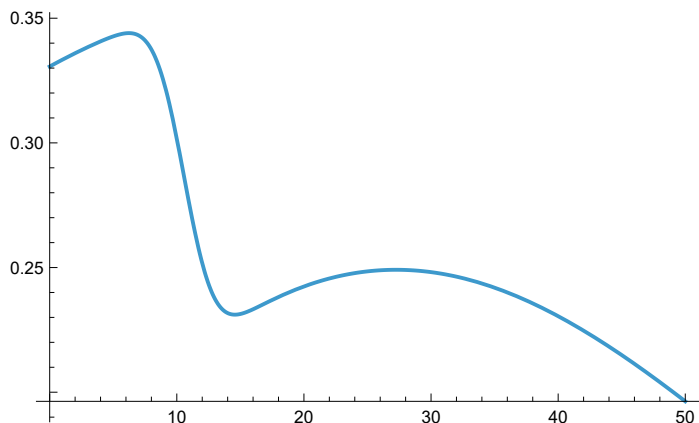
```
In[93]:= Plot[{Eprodshadow[t], Eprodsl[t]}, {t, t0, t1}]
プロット
```

Out[93]=



```
In[94]:= Plot[{sstablesol[t]}, {t, t0, t1}]
プロット
```

Out[94]=

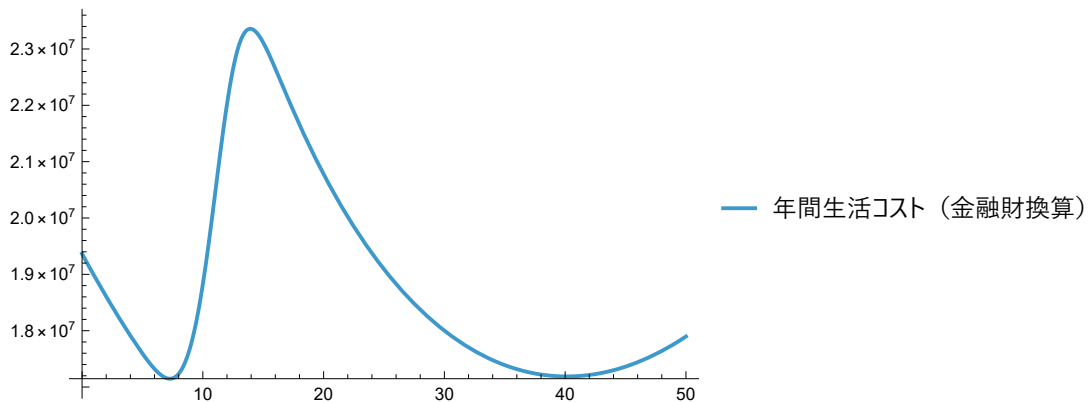


```
In[95]:= Plot[{alpha[t] × Pfsol[t]}, {t, t0, t1}, PlotLegends → {"年間生活コスト（金融財換算）"}]
```

プロット

プロットの凡例

Out[95]=



```
In[96]:= Plot[{Gprod[t] + Gtrans[t], Gwprod[t] + Gwtrans[t]},
```

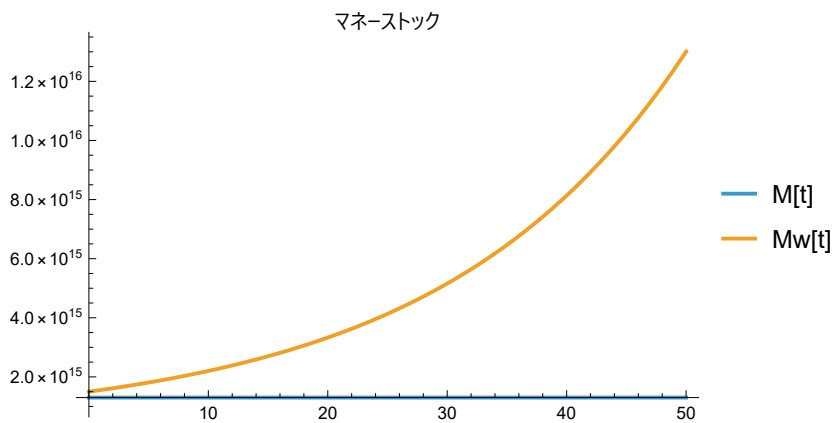
プロット

プロットの凡例

プロットラベル

```
{t, t0, t1}, PlotLegends → {"M[t]", "Mw[t]"}, PlotLabel → "マネーストック"]
```

Out[96]=



```
In[97]:= Plot[{Gprod[t] + Gtrans[t]}, {t, t0, t1},
```

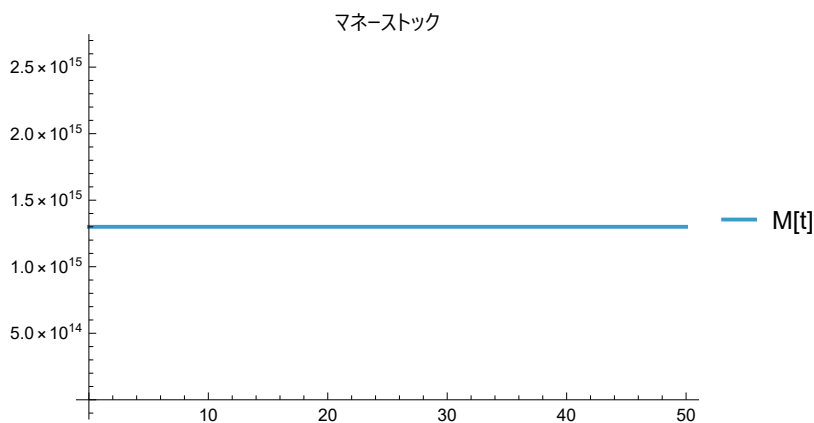
プロット

```
PlotLegends → {"M[t]"}, PlotLabel → "マネーストック"]
```

プロットの凡例

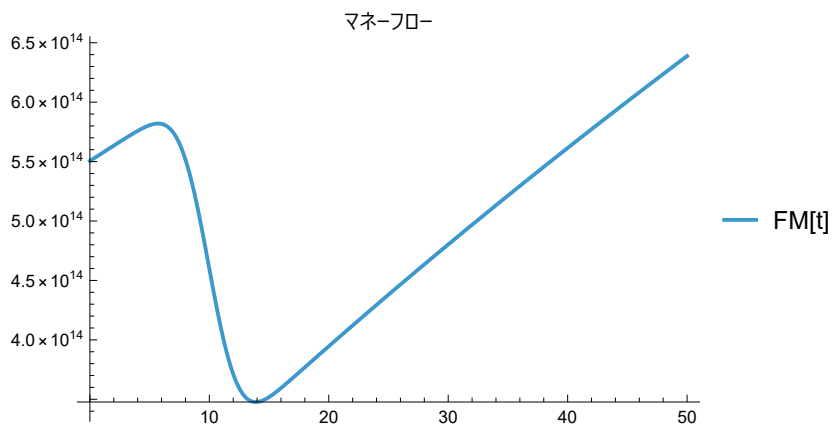
プロットラベル

Out[97]=



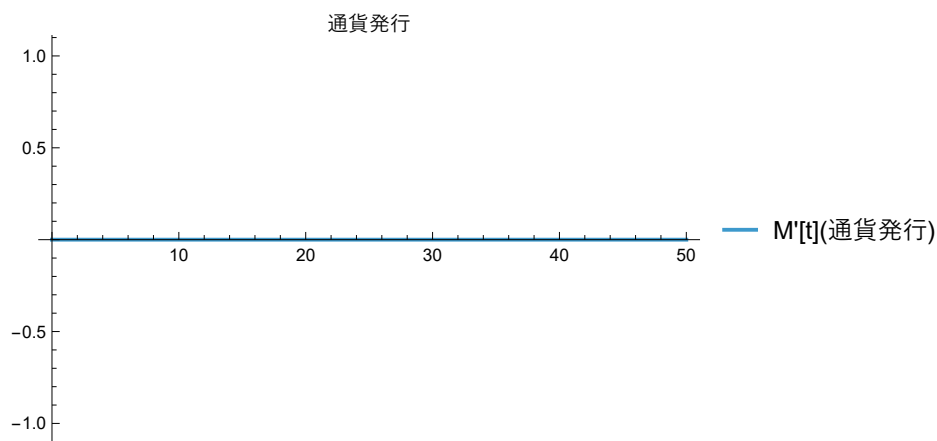
```
In[98]:= Plot[{FMprodsol[t] + FMtranssol[t]}, {t, t0, t1},
  プロット
  PlotLegends → {"FM[t]"}, PlotLabel → "マネーフロー"
  プロットの凡例 プロットラベル
```

Out[98]=



```
In[99]:= Plot[{(Evaluate[dGprod[t] + dGtrans[t]])}, {t, t0, t1},
  プロット 評価
  PlotLegends → {"M'[t] (通貨発行)"}, PlotLabel → "通貨発行"
  プロットの凡例 プロットラベル
```

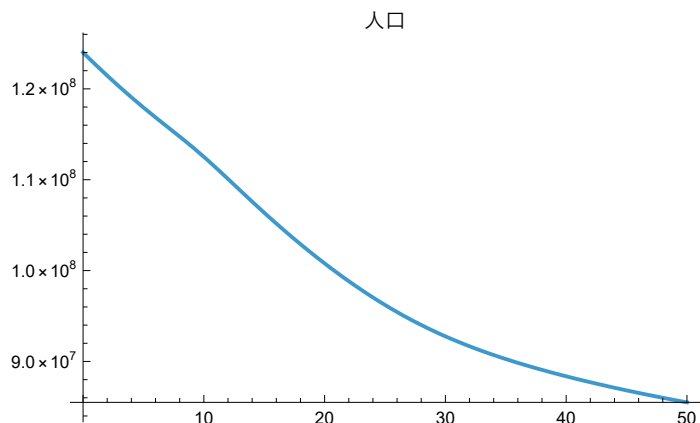
Out[99]=



In[100]:=

```
Plot[{Popusol[t]}, {t, t0, t1}, PlotLabel → "人口"
  プロット プロットラベル
```

Out[100]=



In[101]:=

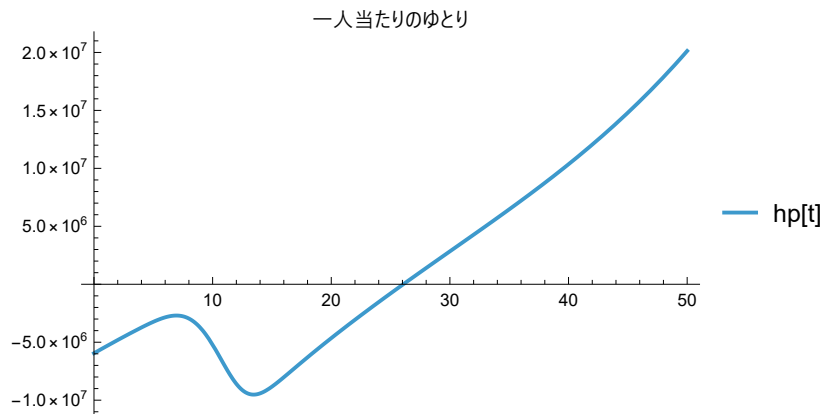
```
Plot[{hpsol[t]}, {t, t0, t1}, PlotLegends → {"hp[t]"}, PlotLabel → "一人当たりのゆとり"]
```

プロット

プロットの凡例

プロットラベル

Out[101]=



In[102]:=

```
Plot[{hrsol[t]}, {t, t0, t1}, PlotLegends → {"hr[t]"}, PlotLabel → "一人当たりのゆとり(比)"]
```

プロット

プロットの凡例

プロットラベル

Out[102]=

