

```

▽ --> /* 6 ツリーモデル */

▽ --> /* [数値処理] → [自動的に数値で出力] */
if numer#false then numer:false else numer:true;
/*浮動小数点の桁数=7に指定 */
fpprec : 7;
/* 浮動小数点表示桁数を7に指定*/
fpprintprec:7;

▽ --> /* 6.1 1 期間モデル */
/* 6.1.2 デリバティブのプライシング */

▽ --> /* Arrow-Debreu 証券の価格 */
arrow_debrue(r,u,d)
:= (phi_u:(r-d)/((1+r)*(u-d)),
    phi_d:(u-r)/((1+r)*(u-d)),[phi_u, phi_d]);

▽ --> arrow_debrue(0.1, 0.2, -0.1);

▽ --> repli(r,u,d,S):= (
/* Arrow-Debreu 証券 phi_u の複製ポートフォリオを求める */
phi_u_S : 1/((u-d) *S) /* 危険資産の保有量 */,
phi_u_B : - (1+d)/((1+r)*(u-d)) /* 安全資産の保有量 */,

/* Arrow-Debreu 証券 phi_d の複製ポートフォリオを求める */
phi_d_S : -1/((u-d) *S) /* 危険資産の保有量 */,
phi_d_B : (1+u)/((1+r)*(u-d))/* 安全資産の保有量 */ ,
matrix(
["phi_u への複製ポートフォリオ", phi_u_S, phi_u_B],
["phi_d への複製ポートフォリオ", phi_d_S, phi_d_B])
);

▽ --> repli (0.1,0.2,-0.1,100);

▽ --> /* リスク中立測度によるプライシング */
mart_deriv(r,u,d,Cu,Cd):=
(
q : (1+r-d) / (u-d) /* リスク中立測度を計算する */,
C0 : (q*Cu + (1-q) *Cd) / (1+r) /* リスク中立測度による期待値を計算 */,
print(" リスク中立測度 q_u=" , q," , " , " デリバティブ価格 =", C0 )
);

▽ --> mart_deriv(0.02,1.1,0.91,50,0);

▽ --> /* 多期間モデル */

```

```

▽ --> deriv_price(r,u,d,N,dt):=
(
  /* 1 期間の推移におけるリスク中立測度を計算する */
  q : (r*dt-d) / (u-d),
  /* (6.17) を計算する */
  /* binomial(N,j)=二項係数を求める関数*/
  C0 :sum(10*j*binomial(N,j) *q^j* (1-q) ^ (N-j)
          / (1+r*dt)^N, j, 0, N),
  print(" マルチンゲール測度 q_u=", q, " ", " ", デリバティブ価格 =", C0 )
);

▽ --> deriv_price (0.02,0.1,-0.09,5,0.2);

▽ --> /*6.2.2 コールオプションのプライシング */
mart_call(r,u,d,S0,K,T,dt):=
(
  /*リスク中立測度を計算する*/
  q : (r*dt-d) / (u-d) ,
  N : T/dt,
  /* truncate () は引数の整数部分を返す関数 */
  N : truncate(N) ,
  C0 :sum(max((1+u) ^j* (1+d) ^ (N-j) *S0-K,0)
          * binomial(N,j) *q^j* (1-q) ^ (N-j) / (1+r*dt)^N, j, 0, N),
  print( " マルチンゲール測度 q_u=", q, " ", " ", " コールオプション価格=", C0 )
);

▽ --> mart_call(0.02,0.1,-0.09,100,80,1,0.02);

▽ --> mart_call_2(r,u,d,S0,K,T,dt):=
(
  N : truncate (T/dt),
  q : (r*dt -d) / (u-d),
  /* 危険資産の下方推移が行使価格を下回らない回数の最小値を求める*/
  a : ceiling(log(K/S0/(1+d)^N )/ log((1+u)/(1+d))),
  /* そのときのコールオプションのペイオフを期待値へと加算していく*/
  tmp : sum( ((1+u) ^j* (1+d) ^ (N-j) *S0-K)
            * binomial(N,j) *q^j* (1-q) ^ (N-j) / (1+r*dt)^N,
            j, a, N),
  print(" マルチンゲール測度 q_u=",q, " ", " ", " コールオプション価格 =", tmp )
);

▽ --> mart_call_2(0.02,0.1,-0.09,100,80,1,0.02);

```