

```
▽ --> /* 5 金利スワップと割引係数 */  
  
▽ --> /* [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;  
  
▽ --> /* 5.2 金利スワップのプライシングと割引係数の導出 */  
  
▽ --> T:[0.5,1,1.5,2,3,4,5,6,7,8,9,10,12,15,20,25,30] /* スワップ年限 */;  
  
▽ --> Swap:[0.345,0.349,0.353,0.359,0.378,0.416,0.479,0.567,0.674,0.795,0.916,  
1.031,1.238,1.479,1.716,1.811,1.853]  
/* スワップ金利 %*/;  
  
▽ --> /* interpolパッケージ内の関数  
cspline(points) 三次スプライン法による多項式内挿  
引数 pointsは以下のいずれか  
2列行列, matrix([2,4],[5,6],[9,3]),  
対のリスト, [[2,4],[5,6],[9,3]],  
数のリスト, [4,6,3], この場合、横座標は自動的に1, 2, 3などに割り当てる. */  
  
▽ --> T_Swap: makelist([T[i], Swap[i]], i, 1, length(T))  
/* [年限, スワップ金利]リスト*/;  
  
▽ --> load(interpol);  
cspline(T_Swap);  
  
▽ --> /* 一つ前の計算結果の式を関数として代入*/  
Yield(x):="%";  
  
▽ --> T1:makelist(0.5*i, i, 1, 60) /* 0.5 年毎に 30 年まで年限を設定 */;  
  
▽ --> Yield(T1) /* 補間されたswap金利の値 */;
```

```
▽ --> load(draw);
      draw2d(
        yrange=[0.2,2],
        point_size=3, point_type=circle,
        /* 以下の点の大きさを3倍, 点のタイプを円に設定 */
        key="Swap Rate",
        points(T_Swap),
        key="Interpolated value",
        explicit(Yield(x), x, 0, 30),
        xlabel="T", ylabel="Swap",
        title="Swap Rates and the interpolated values"
      );

▽ --> print("T=17, ", "Rate=" ,Yield(17)) /* 17年のスワップ金利 */;

▽ --> Par: makelist(Yield(T1[i])/100, i, 1, 60) /* %表示->小数表示へ変換 */;

▽ --> DF: makelist(0, i, 1, 60) /* 割引係数を格納 DF を0で初期化 */;

▽ --> DF[1]:1/ (1+0.5*Par[1]) /* #0.5年のパーレートから割引係数 D_1 を計算*/;

▽ --> for i:2 thru 60 do DF[i] : (1-Par[i]*sum (0.5*DF[j], j, 1, i-1))/ (1+0.5*Par[i])
      /* 1年~30年の割引係数計算 */;

▽ --> T1_DF: makelist([T1[i], DF[i]], i, 1,60)
      /* 描画用[T1,DF]データリスト */;

▽ --> load(draw);
      draw2d(
        point_type=circle,
        points(T1,DF),
        xlabel="T1", ylabel="DF",
        yrange=[0.5,1],
        title="Discount Factors"
      );

▽ --> Spot:makelist((1/DF[i])^(1/T1[i])-1, i, 1, length(T1))
      /* スポット・レート */;

▽ --> /*リストDFからDF[60]を削除*/
      DF1:delete(DF[60], DF);

▽ --> /*リストDF1の先頭に1を加える*/
      DF1: push(1, DF1);
```

```

▽ --> FWD:(DF1/DF-1)/0.5 /*フォワード・レート */;

▽ --> Time_Rate:matrix(T1, 100*Par, 100*Spot, 100*FWD);

▽ --> load(draw);
draw2d(
key="Par", points_joined=true,
points(transpose(submatrix(3,4,Time_Rate)))
/*
submatrix (i_1, ..., i_m, M, j_1, ..., j_n)
submatrix (i_1, ..., i_m, M)
submatrix (M, j_1, ..., j_n)
行列 Mから 行 i_1, ..., i_mを取り除き、
列 j_1, ..., j_nを取り除いた新しい行列を返す
*/
key="Spot", point_type=circle,
points(transpose(submatrix(2,4,Time_Rate))),
key="Foward", point_type=square,
points(transpose(submatrix(2,3,Time_Rate))),
xlabel="T", ylabel="Rate(%)", yrange=[0,3],
title="Par Rates, Spot Rates, and Foward Rates"
);

▽ --> /* 金利スワップの評価とリスク分析 */

▽ --> R_PV: 10000*0.5*0.0045*sum(DF[i], i,1, 9) /* 受け取り額の現在価値 */;

▽ --> P_PV : 10000 * (1-DF[9]) /* 支払い額の現在価値 */;

▽ --> PV:R_PV-P_PV /* 保有スワップの現在価値 */;

▽ --> Swap_p : Swap + 0.01 /* +1bp 変化させた金利カーブ(単位は % ) */;

▽ --> Swap_m : Swap - 0.01 /* -1bp 変化させた金利カーブ(単位は % ) */;

▽ --> cspline(transpose(matrix(T,Swap_p))) /* それぞれのカーブをスプライン補間 */;
sp_p(x):="%";
cspline(transpose(matrix(T,Swap_m)));
sp_m(x):="%";

▽ --> Par_p: makelist(sp_p(T1[i])/100, i, 1, 60);
Par_m: makelist(sp_m(T1[i])/100, i, 1, 60);

```

```
▽ --> /* それぞれのカーブに対する割引係数の導出 */
      DF_p : makelist(0, i, 1, 60);
      DF_m : makelist(0, i, 1, 60);
      DF_p[1] : 1/(1+0.5*Par_p[1]);
      DF_m[1] : 1/(1+0.5*Par_m[1]);
      for i:2 thru 60 do
        DF_p[i] : (1-0.5*Par_p[i]*sum(DF_p[j], j, 1, i-1))/(1+0.5*Par_p[i]);
      for i:2 thru 60 do
        DF_m[i] : (1-0.5*Par_m[i]*sum(DF_m[j], j, 1, i-1))/(1+0.5*Par_m[i]);

▽ --> /* 金利を上下に 1bp 変動させた場合の現在価値を計算 */
      PV_p : 10000 * (0.5 * 0.0045 * sum(DF_p[i], i, 1, 9)- (1-DF_p[9]));
      PV_m : 10000 * (0.5 * 0.0045 * sum(DF_m[i], i, 1, 9)- (1-DF_m[9]));

▽ --> Del : (PV_p-PV_m)/(2*0.0001) /* デルタ */;

▽ --> Gam : (PV_p-2*PV+PV_m)/0.0001^2 /* ガンマ */;

▽ --> PV + Del * 0.001 + 0.5 * Gam * 0.001^2
      /* 金利が 10bp 上昇した場合の現在価値の近似値 */;
```