

## [Help](#)

```
#include "
href../../../../mod/bs1d/bs1d_doublim/bs1d_doublim_h_src.pdfbs1d_doublim.h"
#define INC 1.0e-5 /*Relative Increment for Delta-Hedging*/

static int PutIn_KunitomoIkeda_91(double s, NumFunc_1 *L, NumFunc_1 *U, NumFunc_1 *D,
double t, double r, double divid, double sigma)
{
    double price, delta, out_price, out_delta, price_plus, price_minus;

    pnl_cf_put_bs(s, PayOff->Par[0].Val.V_PDOUBLE, t, r, divid, sigma, &price, &delta);
    PutOut_KunitomoIkeda_91(s, L, U, Rebate, PayOff, t, r, divid, sigma, &out_price, &out_delta);

    /*Price*/
    *ptprice = price - out_price;

    pnl_cf_put_bs(s * (1. + INC), PayOff->Par[0].Val.V_PDOUBLE, t, r, divid, sigma, &price_plus, &delta_plus);
    PutOut_KunitomoIkeda_91(s * (1. + INC), L, U, Rebate, PayOff, t, r, divid, sigma, &out_price_plus, &out_delta_plus);
    price_plus = price - out_price_plus;

    pnl_cf_put_bs(s * (1. - INC), PayOff->Par[0].Val.V_PDOUBLE, t, r, divid, sigma, &price_minus, &delta_minus);
    PutOut_KunitomoIkeda_91(s * (1. - INC), L, U, Rebate, PayOff, t, r, divid, sigma, &out_price_minus, &out_delta_minus);
    price_minus = price - out_price_minus;

    /*Delta*/
    *ptdelta = (price_plus - price_minus) / (2.*s * INC);

    return OK;
}

int CALC(CF_PutIn_KunitomoIkeda)(void *Opt, void *Mod, PricingMethod *Met)
{
    TYPEOPT *ptOpt = (TYPEOPT *)Opt;
    TYPEMOD *ptMod = (TYPEMOD *)Mod;
    double r, divid;

    r = log(1. + ptMod->R.Val.V_DOUBLE / 100.);
    divid = log(1. + ptMod->Divid.Val.V_DOUBLE / 100.);
```

```

    return PutIn_KunitomoIkeda_91(ptMod->S0.Val.V_PDOUBLE, ptOpt->LowerLimit.Val
                                ptOpt->Rebate.Val.V_NUMFUNC_1, ptOpt->PayOff.Val
                                r, divid, ptMod->Sigma.Val.V_PDOUBLE, &(Met->Re
}

static int CHK_OPT(CF_PutIn_KunitomoIkeda)(void *Opt, void *Mod)
{
    Option *ptOpt = (Option *)Opt;
    TYPEOPT *opt = (TYPEOPT *) (ptOpt->TypeOpt);

    if ((opt->Parisian).Val.V_BOOL == FALSE)
        if ((opt->RebOrNo).Val.V_BOOL == NOREBATE)

        return strcmp(((Option *)Opt)->Name, "DoublePutInEuro");
    return WRONG;
}

static int MET(Init)(PricingMethod *Met, Option *Opt)
{
    if (Met->init == 0)
    {
        Met->init = 1;
    }
    return OK;
}

PricingMethod MET(CF_PutIn_KunitomoIkeda) =
{
    "CF_PutIn_KunitomoIkeda",
    {" ", PREMIA_NULLTYPE, {0}, FORBID}},
    CALC(CF_PutIn_KunitomoIkeda),
    {"Price", DOUBLE, {100}, FORBID}, {"Delta", DOUBLE, {100}, FORBID} , {" ", PR
    CHK_OPT(CF_PutIn_KunitomoIkeda),
    CHK_ok,
    MET(Init)
} ;

```