

[Help](#)

```
extern "C" {
#include "
href../../mod/kou1d/kou1d_pad/kou1d_pad_h_src.pdfkou1d_pad.h"
}
#include "
href../../common/math/kirkby/ap_asian_proj_h_src.pdfmath/kirkby/ap_asian_proj

extern "C" {
#if defined(PremiaCurrentVersion) && PremiaCurrentVersion < (2020+2) //The "#els
    static int CHK_OPT(AP_KIRKBY_ASIAN_KOU)(void *Opt, void *Mod)
    {
        return NONACTIVE;
    }
    int CALC(AP_KIRKBY_ASIAN_KOU)(void *Opt, void *Mod, PricingMethod *Met)
    {
        return AVAILABLE_IN_FULL_PREMIA;
    }
#else
    static int Kou_Asian_Kirkby(double Spot, double Strike, NumFunc_2 *po, double
    double lambda, double lambdap, double lambdam, double P, int M0, int L1, int L
    {
int ifCall=0;

    if ((po->Compute) == &Call_OverSpot2)
        ifCall=1;

    Kirkby_PROJ_kou_asian(ifCall, Spot, sigma, lambda, lambdap, lambdam, P, r, divi

        return OK;
    }

int CALC(AP_KIRKBY_ASIAN_KOU)(void *Opt, void *Mod, PricingMethod *Met)
{
    TYPEOPT *ptOpt = (TYPEOPT *)Opt;
    TYPEMOD *ptMod = (TYPEMOD *)Mod;

    int return_value;
    double r, divid, time_spent, pseudo_spot, pseudo_strike;
    double t_0, T_0;
```

```

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

T_0 = ptMod->T.Val.V_DATE;
t_0 = (ptOpt->PathDep.Val.V_NUMFUNC_2)->Par[0].Val.V_PDOUBLE;

if (T_0 < t_0)
{
    Fprintf(TOSCREEN, "T_0 < t_0, untreated case\ n\ n\ n");
    return_value = WRONG;
}
/* Case t_0 <= T_0 */
else
{
    time_spent = (ptMod->T.Val.V_DATE - (ptOpt->PathDep.Val.V_NUMFUNC_2)->Par[0].Val.V_PDOUBLE) / (ptMod->S0.Val.V_PDOUBLE - ptMod->S0.Val.V_PDOUBLE);
    pseudo_spot = (1. - time_spent) * ptMod->S0.Val.V_PDOUBLE;
    pseudo_strike = (ptOpt->PayOff.Val.V_NUMFUNC_2)->Par[0].Val.V_PDOUBLE - ptOpt->PayOff.Val.V_PDOUBLE;

    return_value = Kou_Asian_Kirkby(pseudo_spot, pseudo_strike, ptOpt->PayOff.Val.V_PDOUBLE);
}

return return_value;
}

static int CHK_OPT(AP_KIRKBY_ASIAN_KOU)(void *Opt, void *Mod)
{
    if ((strcmp(((Option *)Opt)->Name, "AsianCallFixedEuro") == 0) || (strcmp(((Option *)Opt)->Name, "AsianPutFixedEuro") == 0))
        return OK;
    return WRONG;
}

#endif //PremiaCurrentVersion

static int MET(Init)(PricingMethod *Met, Option *Mod)
{
    if (Met->init == 0)
    {
        Met->init = 1;
        Met->Par[0].Val.V_PINT = 52;
    }
}

```

```

        Met->Par[1].Val.V_PINT = 10;
Met->Par[2].Val.V_PINT = 9;
    }
    return OK;
}

PricingMethod MET(AP_KIRKBY_ASIAN_KOU) =
{
    "AP_KIRKBY_ASIAN_KOU",
    {
        {"Number of discrete monitoring points", INT, {100}, ALLOW}, {"Grid scale
    },
    CALC(AP_KIRKBY_ASIAN_KOU),
    {{"Price", DOUBLE, {100}, FORBID}, {"Delta", DOUBLE, {100}, FORBID}, {" ", P
    CHK_OPT(AP_KIRKBY_ASIAN_KOU),
    CHK_ok,
    MET(Init)
    } ;
}

```