

[Help](#)

```
extern "C" {
#include "
href../../../../mod/kou1d/kou1d_std/kou1d_std_h_src.pdfkou1d_std.h"
}
#include "
href../../../../common/math/kirkby/ap_bermudan_proj_h_src.pdfmath/kirkby/ap_bermuda
#include "
href../../../../common/math/kirkby/ap_european_proj_h_src.pdfmath/kirkby/ap_europea
#include "
href../../../../common/math/kirkby/model_proj_h_src.pdfmath/kirkby/model_proj.h"

extern "C" {
#if defined(PremiaCurrentVersion) && PremiaCurrentVersion < (2020+2) //The "#els
    static int CHK_OPT(AP_KIRKBY_KOU)(void *Opt, void *Mod)
    {
        return NONACTIVE;
    }
    int CALC(AP_KIRKBY_KOU)(void *Opt, void *Mod, PricingMethod *Met)
    {
        return AVAILABLE_IN_FULL_PREMIA;
    }
#else
    static int Kou_Kirkby(int ifCall,int am,NumFunc_1 *Payoff, double Spot, double
    {

        double Strike;
double price,delta;
        Strike = Payoff->Par[0].Val.V_DOUBLE;

if(am)
    Kirkby_PROJ_kou_amerput(Spot, sigma, lambda, lambdap, lambdam, P, r, divid, T
else
    Kirkby_PROJ_kou_european(ifCall, Spot, sigma, lambda, lambdap, lambdam, P, r,

*ptprice=price;

    return OK;
}
```

```

int CALC(AP_KIRKBY_KOU)(void *Opt, void *Mod, PricingMethod *Met)
{
    TYPEOPT *ptOpt = (TYPEOPT *)Opt;
    TYPEMOD *ptMod = (TYPEMOD *)Mod;
    double r, divid;
int ifCall;
NumFunc_1 *p;

    r = log(1. + ptMod->R.Val.V_DOUBLE / 100.);
    divid = log(1. + ptMod->Divid.Val.V_DOUBLE / 100.);
p = ptOpt->PayOff.Val.V_NUMFUNC_1;

    ifCall = ((p->Compute) == &Call);
    return Kou_Kirkby(ifCall,ptOpt->EuOrAm.Val.V_BOOL,ptOpt->PayOff.Val.V_NUMFU
}

static int CHK_OPT(AP_KIRKBY_KOU)(void *Opt, void *Mod)
{
    if ((strcmp(((Option *)Opt)->Name, "CallEuro") == 0) || (strcmp(((Option *)O
        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 = 14;
Met->Par[2].Val.V_PINT = 14;
    }
    return OK;
}

PricingMethod MET(AP_KIRKBY_KOU) =
{
    "AP_KIRKBY_KOU",
    {

```

```

        {"Number of discrete monitoring points", INT, {100}, ALLOW}, {"Grid scale
    },
    CALC(AP_KIRKBY_KOU),
    {"Price", DOUBLE, {100}, FORBID}, {" ", PREMIA_NULLTYPE, {0}, FORBID}},
    CHK_OPT(AP_KIRKBY_KOU),
    CHK_ok,
    MET(Init)
} ;
}

```