

Help

```

#include "merhes1d_std.h"
#include "math/equity_pricer/levy_diffusion.h"
#include "math/equity_pricer/carr.h"

#if defined(PremiaCurrentVersion) && PremiaCurrentVersion <
    (2010+2) //The "#else" part of the code will be freely av
    ailable after the (year of creation of this file + 2)
static int CHK_OPT(CF_AttariMertonHeston)(void *Opt, void *
    Mod)
{
    return NONACTIVE;
}
int CALC(CF_AttariMertonHeston)(void*Opt,void *Mod,Pricing
    Method *Met)
{
    return AVAILABLE_IN_FULL_PREMIA;
}
#else
int CALC(CF_AttariMertonHeston)(void *Opt, void *Mod, Prici
    ngMethod *Met)
{
    TYPEOPT* ptOpt=(TYPEOPT*)Opt;
    TYPEMOD* ptMod=(TYPEMOD*)Mod;
    NumFunc_1 *p;
    int option_type;
    int std=1;
    if(ptMod->Sigma.Val.V_PDOUBLE==0.0)
    {
        Fprintf(TOSCREEN,"BLACK-SHOLES MODEL{n{n{n"});
        return WRONG;
    }
    else
    {
        double drift;
        Option_Eqd *op;
        Bates_diffusion *Process= Bates_diffusion_create(pt
            Mod->LongRunVariance.Val.V_PDOUBLE,
            pt
            Mod->MeanReversion.hal.V_PDOUBLE,
            pt

```

```

Mod->Rho.Val.V_PDOUBLE,
                                                                    pt
Mod->Sigma.Val.V_PDOUBLE,
                                                                    sq
rt(ptMod->Sigma0.Val.V_PDOUBLE),
                                                                    pt
Mod->Mean.Val.V_PDOUBLE,
                                                                    sq
rt(ptMod->Variance.Val.V_PDOUBLE),
                                                                    pt
Mod->Lambda.Val.V_PDOUBLE,
                                                                    &dr
ift);
    Levy_diffusion * Levy =Levy_diffusion_create(Process,
&Bates_diffusion_characteristic_exponent,&Bates_diffusion_
ln_characteristic_function);
    p=ptOpt->PayOff.Val.V_NUMFUNC_1;
    if ((p->Compute)==&Call)
        option_type=1;
    else
        if((p->Compute)==&Put)
            option_type=2;
        else
            option_type=3;

    op=option_eqd_create(ptOpt->EuOrAm.Val.V_BOOL,option_
type,std,ptMod->S0.Val.V_PDOUBLE,p->Par[0].Val.V_DOUBLE,pt
Opt->Maturity.Val.V_DATE-ptMod->T.Val.V_DATE,0,0);
    option_eqd_set_rate(op,log(1.+ptMod->R.Val.V_DOUBLE/1
00.),log(1.+ptMod->Divid.Val.V_DOUBLE/100.));

    AttariMethod_Vanilla_option_LD(op,0.1,Levy);
    (Met->Res[0].Val.V_DOUBLE)=op->price;
    (Met->Res[1].Val.V_DOUBLE)=op->delta;
    free(op);
    free(Levy);
    free(Process);
    return OK;
}
}

```

```

static int CHK_OPT(CF_AttariMertonHeston)(void *Opt, void *
    Mod)
{
    if ((strcmp( ((Option*)Opt)->Name,"CallEuro")==0)|| (strcmp(
        mp( ((Option*)Opt)->Name,"PutEuro")==0))
        return OK;

    return  WRONG;
}

#endif //PremiaCurrentVersion

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

    return OK;
}

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

```

References