

[Help](#)

```

#include <stdlib.h>
#include <math.h>
#include "pnl/pnl_vector.h"
#include "pnl/pnl_fft.h"
#include "math/wienerhopf.h"
#include "cgmy1d_stdr.h"

#include "pnl/pnl_cdf.h"
#include "pnl/pnl_random.h"
#include "pnl/pnl_specfun.h"

#if defined(PremiaCurrentVersion) && PremiaCurrentVersion < (2012+2) //The "#els
static int CHK_OPT(AP_VAR_FFT)(void *Opt, void *Mod)
{
    return NONACTIVE;
}
int CALC(AP_VAR_FFT)(void *Opt, void *Mod, PricingMethod *Met)
{
    return AVAILABLE_IN_FULL_PREMIA;
}
#else
/*////////////////////////////////////*/

//=====

static int ap_cgmy_var_fft(double alpha, double Spot, NumFunc_1 *p, double T,
                           double mu, double C, double G, double M, double Y, do

{

    double cnu, lp1, lm1, ptvar1, ptcte1;
    double Strike;

    Strike= p->Par[0].Val.V_DOUBLE;

    lm1 = -M;
    lp1 = G;
    cnu = C * pnl_tgamma(-Y);

```

```

var_fft(1, mu, Spot, lm1, lp1,
        Y, Y, cnu, cnu, T, h, Strike, er, alpha, &ptvar1, &ptcte1);

//VaR
*ptvar = ptvar1;
//CTE
*ptcte = ptcte1;

return OK;
}

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

    return ap_cgmy_var_fft(ptOpt->Alpha.Val.V_RGDOUBLE, ptMod->SO.Val.V_PDOUBLE,
                           ptOpt->Maturity.Val.V_DATE - ptMod->T.Val.V_DATE,
                           ptMod->Mu.Val.V_DOUBLE, ptMod->C.Val.V_PDOUBLE, ptMod->
                           Met->Par[1].Val.V_SPDOUBLE, Met->Par[0].Val.V_SPDOUBLE,
                           &(Met->Res[0].Val.V_DOUBLE), &(Met->Res[1].Val.V_DOUBL
}

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

#endif //PremiaCurrentVersion
static int MET(Init)(PricingMethod *Met, Option *Opt)
{
    static int first = 1;

    if (first)
    {
        Met->HelpFilenameHint = "AP_CGMY_VAR";
    }
}

```

```
Met->Par[0].Val.V_PDOUBLE = 2.0;
Met->Par[1].Val.V_PDOUBLE = 0.0001;

    first = 0;
}
return OK;
}

PricingMethod MET(AP_VAR_FFT) =
{
    "AP_VAR_FFT",
    { {"Scale of logprice range", DOUBLE, {100}, ALLOW},
      {"Space Discretization Step", DOUBLE, {500}, ALLOW},
      {" ", PREMIA_NULLTYPE, {0}, FORBID}
    },
    CALC(AP_VAR_FFT),
    { {"Value At Risk", DOUBLE, {100}, FORBID},
      {"Conditional Tail Expectation ", DOUBLE, {100}, FORBID},
      {" ", PREMIA_NULLTYPE, {0}, FORBID}
    },
    CHK_OPT(AP_VAR_FFT),
    CHK_split,
    MET(Init)
};
```