```
  Help
/*----------------------------------------------------------
    -----*/
/*   Monte Carlo algorithm for caplet prices in one-factor
    LMM with jumps  */
/*   Algorithm of Glasserman/Merener
         */
/*
         */
/*----------------------------------------------------------
    -----*/
/*  Sonke Blunck, Premia 2005
         */
/*----------------------------------------------------------
    -----*/

#include  "glassermanmerener.h"

extern "C"{
#include  "lmm_jump1d_stdi.h"
#include "enums.h"

#if defined(PremiaCurrentVersion) && PremiaCurrentVersion <
     (2007+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(MC_GM)(void *Opt, void *Mod)
{
  return NONACTIVE;
}
int CALC(MC_GM)(void *Opt,void *Mod,PricingMethod *Met)
{
  return AVAILABLE_IN_FULL_PREMIA;
}
#else

static int mc_glassermanmerenenr_caplet(NumFunc_1 *p,
    double l0,double t0, double sigma,double capletMat ,double stri
    ke,double tenor, int generator,long numberMCPaths,double *
    price)
{
  capletMat=capletMat-t0;

  return lmm_jump_caplet_MC_pricer(tenor,capletMat,strike,
    l0,sigma,numberMCPaths,generator, price);
}

int CALC(MC_GM)(void *Opt,void *Mod,PricingMethod *Met)
{
  TYPEOPT* ptOpt=(TYPEOPT*)Opt;
  TYPEMOD* ptMod=(TYPEMOD*)Mod;
  int init_mc;
  init_mc= pnl_rand_init(Met->Par[0].Val.V_ENUM.value,1,
    Met->Par[1].Val.V_LONG);
  if (init_mc != OK) return init_mc;
  return mc_glassermanmerenenr_caplet(ptOpt->PayOff.Val.V_
    NUMFUNC_1,ptMod->l0.Val.V_PDOUBLE,
                                      ptMod->T.Val.V_DATE,
                                      ptMod->Sigma.Val.V_
    PDOUBLE,
                                      ptOpt->BMaturity.Val.
    V_DATE,
                                      ptOpt->FixedRate.Val.
    V_PDOUBLE,
                                      ptOpt->ResetPeriod.
    Val.V_DATE,
                                      Met->Par[0].Val.V_
    ENUM.value,
                                      Met->Par[1].Val.V_LON
    G,
                                      &(Met->Res[0].Val.V_
    DOUBLE));

}
static int CHK_OPT(MC_GM)(void *Opt, void *Mod)
{

  if ((strcmp(((Option*)Opt)->Name,"Caplet")==0))
    return OK;
  else
    return WRONG;
}
```

```
#endif //PremiaCurrentVersion
static int MET(Init)(PricingMethod *Met,Option *Opt)
{
  if ( Met->init == 0)
    {
      Met->init=1;

      Met->Par[0].Val.V_ENUM.value=0;
      Met->Par[0].Val.V_ENUM.members=&PremiaEnumRNGs;
      Met->Par[1].Val.V_LONG=100;
    }
  return OK;
}

PricingMethod MET(MC_GM)=
{
  "MC_GlassermanMerener",
  {{"RandomGenerator",ENUM,{100},ALLOW},
   {"N Simulation",LONG,{100},ALLOW},
   {" ",PREMIA_NULLTYPE,{0},FORBID}},
  CALC(MC_GM),
  {{"Price",DOUBLE,{100},FORBID}/*,{"Delta",DOUBLE,{100},FO
    RBID}*/ ,{" ",PREMIA_NULLTYPE,{0},FORBID}},
  CHK_OPT(MC_GM),
  CHK_ok,
  MET(Init)
} ;
}
```

# References