```
   Help
extern "C"{
#include "temperedstable1d_vol.h"
}
#include "math/numerics.h"
extern "C"{

#if defined(PremiaCurrentVersion) && PremiaCurrentVersion <
     (2008+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(AP_REPL2_VARIANCESWAP)(void *Opt, void *
    Mod)
{
  return NONACTIVE;
}
int CALC(AP_REPL2_VARIANCESWAP)(void *Opt,void *Mod,Pricing
    Method *Met)
{
return AVAILABLE_IN_FULL_PREMIA;
}
#else

  static double replFun(double v, double m);

  //---------------------------------------------------------
    ---------------
  static int ap_cgmy_varswap_repl2(double S0, double Strike
    , double T, double r, double divid, double ap, double am,
    double lap,double lam,double cpp,double cmm, double *fairval,
    double *ptprice)
  {
    //S0 is a forward price
  double *replStrikes;
  double *replOptions;
  double *replWeights;
  int *CallPuts;
  int flag;
  double strikestep=0.05*S0, kfirst=0.5*S0;
  double pvfactor=exp(-r*T);

  int k, k0, res, replN=22;
```

```
double optprice, tweight, tstrike, tprice;

replStrikes = new double[replN];
replOptions = new double[replN];
replWeights = new double[replN];
CallPuts = new int[replN];

tprice=0.0;

tstrike=S0;
k=0;
flag=1;
while((k<replN)&&(flag))
{
  replStrikes[k]=kfirst+k*strikestep;
  CallPuts[k]=(S0<=replStrikes[k]);
  flag=!CallPuts[k];
  k++;
}

if (S0==replStrikes[k-1]) {
  replStrikes[k-1]+=strikestep;
  kfirst+=strikestep;}

k0=k-2;
for(;k<replN;k++)
{
  replStrikes[k]=kfirst+k*strikestep;
  CallPuts[k]=1;
}

//weights for puts
tweight=0;
tstrike=S0;
for(k=k0;k>=0;k--)
{
  replWeights[k]=( replFun(replStrikes[k], S0)-replFun(
  tstrike, S0) ) /strikestep - tweight;
  tweight+= replWeights[k];
  res=iac_kobol_europut(CallPuts[k], lam, lap, am, ap,
  cmm, cpp, r, T, tstrike, S0*pvfactor, 0.00000001, &optpric
  e);
  if(res) {return 1;}
  replOptions[k]=optprice;
  tstrike = replStrikes[k];
  tprice += replOptions[k]*replWeights[k];
}
//weights for calls
tweight=0;
tstrike=S0;
for(k=k0+1;k<replN;k++)
{
  replWeights[k]=( replFun(replStrikes[k], S0) - replF
  un(tstrike, S0) ) /strikestep - tweight;
  tweight+= replWeights[k];
  res=iac_kobol_europut(CallPuts[k], lam, lap, am, ap,
  cmm, cpp, r, T, tstrike, S0*pvfactor, 0.00000001, &optpric
  e);
  if(res) {return 1;}
  replOptions[k]=optprice;
  tstrike = replStrikes[k];
  tprice+= replOptions[k]*replWeights[k];
}

//portfolio value
tprice=2.0/T*(/*1.0+r*T-exp(r*T)+*/tprice);

//fair strike of variance swap, in annual volatility po
  ints
*fairval= sqrt(tprice/pvfactor)*100;
// strike in variance points
kfirst = pvfactor*Strike*Strike;
// price of var swap
*ptprice= tprice*10000-kfirst;

delete [] replStrikes;
delete [] replOptions;
delete [] replWeights;
delete [] CallPuts;

return OK;
}
```

```
 //------------------------------------------------------
  -----------------
static double replFun(double v, double m)
{
  return (v-m)/m-log(v/m);
}

int CALC(AP_REPL2_VARIANCESWAP)(void *Opt,void *Mod,Prici
  ngMethod *Met)
{
  TYPEOPT* ptOpt=(TYPEOPT*)Opt;
  TYPEMOD* ptMod=(TYPEMOD*)Mod;
  double r, divid, strike, spot;
  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;
  strike=p->Par[0].Val.V_DOUBLE;
  spot=ptMod->S0.Val.V_DOUBLE;

  return ap_cgmy_varswap_repl2(
    spot, strike, ptOpt->Maturity.Val.V_DATE-ptMod->T.Val
  .V_DATE, r, divid,  ptMod->AlphaPlus.Val.V_PDOUBLE, ptMod-
  >AlphaMinus.Val.V_PDOUBLE, ptMod->LambdaPlus.Val.V_PDOUB
  LE, ptMod->LambdaMinus.Val.V_PDOUBLE, ptMod->CPlus.Val.V_PDO
  UBLE, ptMod->CMinus.Val.V_PDOUBLE,
  &(Met->Res[0].Val.V_DOUBLE), &(Met->Res[1].Val.V_
  DOUBLE));

}

static int CHK_OPT(AP_REPL2_VARIANCESWAP)(void *Opt, voi
  d *Mod)
{
  if ((strcmp( ((Option*)Opt)->Name,"VarianceSwap")==0))
    return OK;

  return WRONG;
}
```

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

    if (first)
    {
      first=0;
      Met->HelpFilenameHint = "ap_cgmy_varswap_repl2";
    }
    return OK;
  }

  PricingMethod MET(AP_REPL2_VARIANCESWAP)=
  {
    "AP_CGMY_VARSWAP_REP2",
    {{" ",PREMIA_NULLTYPE,{0},FORBID}},
    CALC(AP_REPL2_VARIANCESWAP),
    {   {"Fair strike in annual volatility points",DOUBLE,{
    100},FORBID},
        {"Price in 10000 variance points",DOUBLE,{100},FORB
    ID},
        {" ",PREMIA_NULLTYPE,{0},FORBID}},
    CHK_OPT(AP_REPL2_VARIANCESWAP),
    CHK_ok ,
    MET(Init)
  } ;

  /*////////////////////////////////////*/
}
```

References