

[Help](#)

```
#include "bs2d_std2d.h"

int MOD_OPT(ChkMix)(Option *Opt, Model *Mod)
{
    TYPEOPT *ptOpt = (TYPEOPT *) (Opt->TypeOpt);
    TYPEMOD *ptMod = (TYPEMOD *) (Mod->TypeModel);
    int status = OK;

    if (ptOpt->Maturity.Val.V_DATE <= ptMod->T.Val.V_DATE)
    {
        Fprintf(TOSCREENANDFILE, "Current date greater than maturity!\ n");
        status += 1;
    };

    return status;
}

extern PricingMethod MET(CF_CallMax);
extern PricingMethod MET(CF_Exchange);
extern PricingMethod MET(CF_PutMin);
extern PricingMethod MET(FD_Adi);
extern PricingMethod MET(FD_Explicit);
extern PricingMethod MET(FD_VFExplicit);
extern PricingMethod MET(FD_Howard);
extern PricingMethod MET(FD_Multigrid);
extern PricingMethod MET(FD_FMGH);
extern PricingMethod MET(FD_GMRES);
extern PricingMethod MET(FD_Psor);
extern PricingMethod MET(FD_BCGStab);
extern PricingMethod MET(MC_Standard2D);
extern PricingMethod MET(TR_BoyleEvnineGibbs);
extern PricingMethod MET(TR_KamradRitchken);
extern PricingMethod MET(TR_ProductTR);
extern PricingMethod MET(MC_LongstaffSchwartz2D);
extern PricingMethod MET(MC_RandomQuantization2D);
extern PricingMethod MET(MC_BarraquandMartineau2D);
extern PricingMethod MET(MC_BroadieGlassermann2D);
```

```

extern PricingMethod MET(MC_LionsRegnier2D);
extern PricingMethod MET(MC_BGRS2D);
extern PricingMethod MET(MC_JainOosterlee2D);
/*extern PricingMethod MET(TR_Euler);*/

```

```

PricingMethod *MOD_OPT(methods) [] =
{
    &MET(CF_CallMax),
    &MET(CF_Exchange),
    &MET(CF_PutMin),
    &MET(FD_Adi),
    &MET(FD_Explicit),
    &MET(FD_VFExplicit),
    &MET(FD_Howard),
    &MET(FD_Multigrid),
    &MET(FD_FMGH),
    &MET(FD_GMRES),
    &MET(FD_Psor),
    &MET(FD_BCGStab),
    &MET(MC_Standard2D),
    &MET(TR_BoyleEvnineGibbs),
    &MET(TR_KamradRitchken),
    &MET(TR_ProductTR),
    &MET(MC_LongstaffSchwartz2D),
    &MET(MC_RandomQuantization2D),
    &MET(MC_BarraquandMartineau2D),
    &MET(MC_BroadieGlassermann2D),
    &MET(MC_LionsRegnier2D),
    &MET(MC_BGRS2D),
    &MET(MC_JainOosterlee2D),
    /*&MET(TR_Euler),*/
    NULL
};

```

```

extern DynamicTest MOD_OPT(test);
DynamicTest *MOD_OPT(tests) [] =
{
    &MOD_OPT(test),
    NULL
};

```

```
Pricing MOD_OPT(pricing) =  
{  
    ID_MOD_OPT,  
    MOD_OPT(methods),  
    MOD_OPT(tests),  
    MOD_OPT(ChkMix)  
};
```

```
/* Utility function */
```

```
int MOD_OPT(Delta_Operator)(double u1, double d1, double u2, double d2, double s  
                           double stock2, double puu, double pud, double pdu,  
                           double pdd, double *ptdelta1, double *ptdelta2)  
{  
    *ptdelta1 = ((d2 - 1.) * (pdu - puu) + (u2 - 1.) * (pud - pdd)) / (stock2 * (u  
    *ptdelta2 = ((d1 - 1.) * (pud - puu) + (u1 - 1.) * (pdu - pdd)) / (stock1 * (u  
  
    return OK;  
}
```