

## Help

```

#include "bs1d_std.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 great
        er than maturity!\n");
        status+=1;
    };

    return status;
}

extern PricingMethod MET(CF_Call);
extern PricingMethod MET(CF_Put);
extern PricingMethod MET(CF_CallSpread);
extern PricingMethod MET(CF_Digit);
extern PricingMethod MET(AP_Bjerk SundStensland);
extern PricingMethod MET(AP_BunchJohnsonn);
extern PricingMethod MET(
    AP_HoStapletonSubrahmanyam);
extern PricingMethod MET(AP_McMillan);
extern PricingMethod MET(AP_Whaley);
extern PricingMethod MET(AP_Ju_PutAmer);
extern PricingMethod MET(AP_Carr_PutAmer);
extern PricingMethod MET(AP_Luba_CallAmer);
extern PricingMethod MET(AP_Lba_CallAmer);
extern PricingMethod MET(FD_BrennanSchwartz);
extern PricingMethod MET(FD_Explicit);
extern PricingMethod MET(FD_Gauss);
extern PricingMethod MET(FD_Psor);
extern PricingMethod MET(FD_Cryer);
extern PricingMethod MET(FD_Sor);

```

```

extern PricingMethod MET(FD_Galerkin_Discontinou
    s);
extern PricingMethod MET(FD_Howard_amer1);
extern PricingMethod MET(FD_Multigrid_Euro);
extern PricingMethod MET(FD_FMGH);
extern PricingMethod MET(MC_Standard);
extern PricingMethod MET(MC_Antithetic);
extern PricingMethod MET(TR_ThirdMoment);
extern PricingMethod MET(TR_LnThirdMoment);
extern PricingMethod MET(TR_CoxRossRubinstein);
extern PricingMethod MET(TR_Euler);
extern PricingMethod MET(TR_KamradRitchken);
extern PricingMethod MET(TR_ExtendedCRR);
extern PricingMethod MET(TR_HullWhite);
extern PricingMethod MET(TR_BBSR);
extern PricingMethod MET(TR_Patry);
extern PricingMethod MET(TR_Patry1);
extern PricingMethod MET(MC_Antithetic);
extern PricingMethod MET(MC_LongstaffSchwartz);
extern PricingMethod MET(MC_RandomQuantization);
extern PricingMethod MET(MC_BarraquandMartineau);
extern PricingMethod MET(MC_BroadieGlassermann);
extern PricingMethod MET(MC_TsitsiklisVanRoy);
extern PricingMethod MET(MC_Rogers);
extern PricingMethod MET(MC_LionsRegnier);

PricingMethod* MOD_OPT(methods) []={
    &MET(CF_Call),
    &MET(CF_Put),
    &MET(CF_CallSpread),
    &MET(CF_Digit),
    &MET(AP_Bjerk SundStensland),
    &MET(AP_BunchJohnsonn),
    &MET(AP_HoStapletonSubrahmanyam),
    &MET(AP_McMillan),
    &MET(AP_Whaley),
    &MET(AP_Ju_PutAmer),
    &MET(AP_Carr_PutAmer),
    &MET(AP_Luba_CallAmer),
    &MET(AP_Lba_CallAmer),
    &MET(FD_BrennanSchwartz),

```

```

    &MET(FD_Explicit),
    &MET(FD_Gauss),
    &MET(FD_Psor),
    &MET(FD_Cryer),
    &MET(FD_Sor),
        &MET(FD_Galerkin_Discontinuous),
        &MET(FD_Howard_amer1),
        &MET(FD_Multigrid_Euro),
        &MET(FD_FMGH),
        &MET(MC_Standard),
    &MET(MC_Antithetic),
    &MET(TR_ThirdMoment),
    &MET(TR_LnThirdMoment),
    &MET(TR_CoxRossRubinstein),
    &MET(TR_Euler),
    &MET(TR_KamradRitchken),
    &MET(TR_ExtendedCRR),
    &MET(TR_HullWhite),
    &MET(TR_BBSR),
        &MET(TR_Patry),
        &MET(TR_Patry1),
    &MET(MC_LongstaffSchwartz),
    &MET(MC_RandomQuantization),
    &MET(MC_BarraquandMartineau),
    &MET(MC_BroadieGlassermann),
    &MET(MC_TsitsiklisVanRoy),
        &MET(MC_Rogers),
    &MET(MC_LionsRegnier),
    NULL
};

extern DynamicTest MOD_OPT(test);
extern DynamicTest MOD_OPT(testpatry);
extern DynamicTest MOD_OPT(testpatry1);
extern DynamicTest MOD_OPT(test1);
extern DynamicTest MOD_OPT(test2);
extern DynamicTest MOD_OPT(test3);

DynamicTest* MOD_OPT(tests)[]={
    &MOD_OPT(test),
    &MOD_OPT(testpatry),

```

```
        &MOD_OPT(testpatry1),
        &MOD_OPT(test1),
        &MOD_OPT(test2),
        &MOD_OPT(test3),
        NULL
    };

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

## References