

[Help](#)

```
#include "
href../../../../mod/merhes1d/merhes1d_std/merhes1d_std_h_src.pdfmerhes1d_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 greater than maturity!\ n");
        status += 1;
    };

    return status;
}

extern PricingMethod MET(CF_CallMertonHeston);
extern PricingMethod MET(CF_PutMertonHeston);
extern PricingMethod MET(CF_CarrMertonHeston);
extern PricingMethod MET(CF_AttariMertonHeston);
extern PricingMethod MET(MC_Polynomial);
extern PricingMethod MET(MC_Alfonsi_Bates);
extern PricingMethod MET(MC_HybridTree_Bates);
extern PricingMethod MET(FD_HybridTree_Bates);
extern PricingMethod MET(FD_Adi_Bates);
extern PricingMethod MET(FD_NataliniBriani_MERHES);
extern PricingMethod MET(MC_AM_Alfonsi_LongstaffSchwartz_Bates);
extern PricingMethod MET(MC_AM_Alfonsi_AndersenBroadie_Bates);
extern PricingMethod MET(AP_Alos_Bates);
PricingMethod *MOD_OPT(methods) [] =
{
    &MET(CF_CallMertonHeston),
    &MET(CF_PutMertonHeston),
    &MET(CF_CarrMertonHeston),
    &MET(CF_AttariMertonHeston),
    &MET(MC_Polynomial),
```

```

    &MET(MC_Alfonsi_Bates),
    &MET(MC_HybridTree_Bates),
    &MET(FD_HybridTree_Bates),
    &MET(FD_Adi_Bates),
    &MET(FD_NataliniBriani_MERHES),
    &MET(MC_AM_Alfonsi_LongstaffSchwartz_Bates),
    &MET(MC_AM_Alfonsi_AndersenBroadie_Bates),
    &MET(AP_Alos_Bates),
    NULL
};

DynamicTest *MOD_OPT(tests)[] =
{
    NULL
};

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

```