

[Help](#)

```
#include "
href../../mod/variancegamma1d/variancegamma1d_std/variancegamma1d_std_h_src.p

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(AP_CarrVG);
extern PricingMethod MET(AP_fastwhamer_vg);
extern PricingMethod MET(AP_fastwhamerdig_vg);
extern PricingMethod MET(AP_backwardfourierdig_vg);
extern PricingMethod MET(AP_backwardfourieramer_vg);
extern PricingMethod MET(AP_COSINEFILTER_vg);
extern PricingMethod MET(FD_ImpExp);
extern PricingMethod MET(TR_MSS_VG);

PricingMethod *MOD_OPT(methods) [] =
{
    &MET(AP_fastwhamer_vg),
    &MET(AP_fastwhamerdig_vg),
    &MET(AP_backwardfourierdig_vg),
    &MET(AP_backwardfourieramer_vg),
    &MET(FD_ImpExp),
    &MET(AP_CarrVG),
    &MET(AP_COSINEFILTER_vg),
    &MET(TR_MSS_VG),

    NULL
}
```

```
};

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

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