

[Help](#)

```

#include "bs1d_pad.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 g
reater than maturity!\n");
        status+=1;
    };
    if ((ptOpt->MinOrElse).Val.V_BOOL==MINIMUM)
    {
        if ((ptOpt->PathDep.Val.V_NUMFUNC_2)->
            Par[4].Val.V_PDOUBLE>ptMod->S0.Val.V_PDOUBLE)
        {
            Fprintf(TOSCREENANDFILE,"Minimum g
reater than spot!\n");
            status+=1;
        };
    }
    if ((ptOpt->MinOrElse).Val.V_BOOL==MAXIMUM)
    {
        if ((ptOpt->PathDep.Val.V_NUMFUNC_2)->
            Par[4].Val.V_PDOUBLE<ptMod->S0.Val.V_PDOUBLE)
        {
            Fprintf(TOSCREENANDFILE,"Maximum
lower than spot!\n");
            status+=1;
        };
    }
    return status;
}

extern PricingMethod MET(
    MC_FloatingAsian_Standard);

```

```

extern PricingMethod MET(CF_Fixed_CallLookBack);
extern PricingMethod MET(CF_Fixed_PutLookBack);
extern PricingMethod MET(
    CF_Floating_CallLookBack);
extern PricingMethod MET(
    CF_Floating_PutLookBack);
extern PricingMethod MET(AP_FixedAsian_Laplace);
extern PricingMethod MET(AP_FixedAsian_Levy);
extern PricingMethod MET(
    AP_FixedAsian_TurnbullWakeman);
extern PricingMethod MET(
    AP_FixedAsian_MilevskyPosner);
extern PricingMethod MET(
    AP_FixedAsian_ThompsonLow);
extern PricingMethod MET(
    AP_FixedAsian_ThompsonUp);
extern PricingMethod MET(
    AP_FixedAsian_FusaiTagliani);
extern PricingMethod MET(AP_FixedAsian_Zhang);
extern PricingMethod MET(AP_FloatingAsian_Zhang);
extern PricingMethod MET(
    FD_FixedAsian_RodgerShi);
extern PricingMethod MET(FD_FixedAsian_BenHameurB
    retonLecuyer);
extern PricingMethod MET(
    MC_FixedAsian_KemnaVorst);
extern PricingMethod MET(
    MC_FixedAsian_Glassermann);
extern PricingMethod MET(MC_FixedAsian_RobbinsMon
    ro);
extern PricingMethod MET(
    MC_LookBackMax_Andersen);
extern PricingMethod MET(
    MC_LookBackMin_Andersen);
extern PricingMethod MET(TR_Babbs_Call);
extern PricingMethod MET(TR_Babbs_Put);
extern PricingMethod MET(TR_Asian_FSG);
PricingMethod* MOD_OPT(methods)[]={
    &MET(MC_FloatingAsian_Standard),
    &MET(CF_Fixed_CallLookBack),
    &MET(CF_Fixed_PutLookBack),

```

```

    &MET(CF_Floating_CallLookBack),
    &MET(CF_Floating_PutLookBack),
    &MET(AP_FixedAsian_Laplace),
    &MET(AP_FixedAsian_Levy),
    &MET(AP_FixedAsian_TurnbullWakeman),
    &MET(AP_FixedAsian_MilevskyPosner),
    &MET(AP_FixedAsian_ThompsonLow),
    &MET(AP_FixedAsian_ThompsonUp),
    &MET(AP_FixedAsian_FusaiTagliani),
    &MET(AP_FixedAsian_Zhang),
    &MET(AP_FloatingAsian_Zhang),
    &MET(FD_FixedAsian_RodgerShi),
    &MET(FD_FixedAsian_BenHameurBretonLecuyer),
    &MET(MC_FixedAsian_KemnaVorst),
    &MET(MC_FixedAsian_Glassermann),
    &MET(MC_FixedAsian_RobbinsMonro),
    &MET(MC_LookBackMax_Andersen),
    &MET(MC_LookBackMin_Andersen),
    &MET(TR_Babbs_Call),
    &MET(TR_Babbs_Put),
    &MET(TR_Asian_FSG),
    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)
};

```

References