

[Help](#)

```
#include "lmm_stochvol_piterbarg.h"
#include "premia_obj.h"
#include "chk.h"
#include "model.h"

extern PremiaEnum flat;

double MOD(GetYield)(TYPEMOD *pt)
{
    VAR *Par;
    Par = lookup_premia_enum_par(&(pt->Flag_InitialYieldCurve), 0);
    return Par[0].Val.V_PDOUBLE;
}

char *MOD(GetCurve)(TYPEMOD *pt)
{
    VAR *Par;
    Par = lookup_premia_enum_par(&(pt->Flag_InitialYieldCurve), 1);
    return Par[0].Val.V_FILENAME;
}

static int MOD(Init)(Model *model)
{
    VAR *Par;
    TYPEMOD *pt = (TYPEMOD *) (model->TypeModel);

    if (model->init == 0)
    {
        model->init = 1;
        model->nvar = 0;
        pt->T.Vname = "Current Date";
        pt->T.Vtype = DATE;
        pt->T.Val.V_DATE = 0.0;
        pt->T.Viter = ALLOW;
        model->nvar++;

        pt->Flag_InitialYieldCurve.Vname = "Initial Yield Curve";
        pt->Flag_InitialYieldCurve.Vtype = ENUM;
        pt->Flag_InitialYieldCurve.Val.V_ENUM.value = 0;
        pt->Flag_InitialYieldCurve.Val.V_ENUM.members = &PremiaEnumFlat;
    }
}
```

```
pt->Flag_InitialYieldCurve.Viter = ALLOW;
model->nvar++;
Par = lookup_premia_enum_par(&(pt->Flag_InitialYieldCurve), 0);
Par[0].Vname = "Yield Value";
Par[0].Vtype = PDOUBLE;
Par[0].Val.V_PDOUBLE = 0.05;
Par[0].Viter = FORBID;
Par = lookup_premia_enum_par(&(pt->Flag_InitialYieldCurve), 1);
Par[0].Vname = "Yield Curve";
Par[0].Vtype = FILENAME;
Par[0].Val.V_FILENAME = NULL;
Par[0].Viter = FORBID;

pt->Var_SpeedMeanReversion.Vname = "Variance Speed of Mean Reversion";
pt->Var_SpeedMeanReversion.Vtype = DOUBLE;
pt->Var_SpeedMeanReversion.Val.V_DOUBLE = 2.0;
pt->Var_SpeedMeanReversion.Viter = ALLOW;
model->nvar++;

pt->Var_Volatility.Vname = "Variance Volatility";
pt->Var_Volatility.Vtype = DOUBLE;
pt->Var_Volatility.Val.V_DOUBLE = 0.1;
pt->Var_Volatility.Viter = ALLOW;
model->nvar++;

pt->SkewsParams_a.Vname = "Skews:(a(Tn-t)+b)exp(-c(Tn-t))+d : a";
pt->SkewsParams_a.Vtype = DOUBLE;
pt->SkewsParams_a.Val.V_DOUBLE = 0.1;
pt->SkewsParams_a.Viter = ALLOW;
model->nvar++;

pt->SkewsParams_b.Vname = "b";
pt->SkewsParams_b.Vtype = DOUBLE;
pt->SkewsParams_b.Val.V_DOUBLE = 0.1;
pt->SkewsParams_b.Viter = ALLOW;
model->nvar++;

pt->SkewsParams_c.Vname = "c";
pt->SkewsParams_c.Vtype = DOUBLE;
pt->SkewsParams_c.Val.V_DOUBLE = 0.1;
pt->SkewsParams_c.Viter = ALLOW;
```

```

model->nvar++;

pt->SkewsParams_d.Vname = "d";
pt->SkewsParams_d.Vtype = DOUBLE;
pt->SkewsParams_d.Val.V_DOUBLE = 0.1;
pt->SkewsParams_d.Viter = ALLOW;
model->nvar++;

pt->VolsParams_a.Vname = "Vols:(a(Tn-t)+b)exp(-c(Tn-t))+d : a";
pt->VolsParams_a.Vtype = DOUBLE;
pt->VolsParams_a.Val.V_DOUBLE = 0.1;
pt->VolsParams_a.Viter = ALLOW;
model->nvar++;

pt->VolsParams_b.Vname = "b";
pt->VolsParams_b.Vtype = DOUBLE;
pt->VolsParams_b.Val.V_DOUBLE = 0.1;
pt->VolsParams_b.Viter = ALLOW;
model->nvar++;

pt->VolsParams_c.Vname = "c";
pt->VolsParams_c.Vtype = DOUBLE;
pt->VolsParams_c.Val.V_DOUBLE = 0.1;
pt->VolsParams_c.Viter = ALLOW;
model->nvar++;

pt->VolsParams_d.Vname = "d";
pt->VolsParams_d.Vtype = DOUBLE;
pt->VolsParams_d.Val.V_DOUBLE = 0.1;
pt->VolsParams_d.Viter = ALLOW;
model->nvar++;

}

Par = lookup_premia_enum_par(&(pt->Flag_InitialYieldCurve), 1);
if (Par[0].Val.V_FILENAME == NULL)
{
    if ((Par[0].Val.V_FILENAME = malloc(sizeof(char) * MAX_PATH_LEN)) == NULL)
        return MEMORY_ALLOCATION_FAILURE;
    sprintf(Par[0].Val.V_FILENAME, "%s%sinitialyield.dat", premia_data_dir, pa
}

```

```
    return OK;
}

TYPEMOD Lmm_StochVol_Piterbarg;
MAKEMOD(Lmm_StochVol_Piterbarg);
```