

Help

```

#include "hes1d.h"

extern char* path_sep;
extern char **error_msg;

int MOD(Get)(int user,Planning *pt_plan,Model *
    model)
{
    TYPEMOD* pt=(TYPEMOD*)(model->TypeModel);

    MOD(Init)(model);

    if (user==TOSCREEN)
        if ((model->Show)(user,pt_plan,model))
            do
            {
                Fprintf(TOSCREEN,"-----
                -----Model:%s{n",model->Name);

                ScanVar(pt_plan,user,&(pt->T));
                ScanVar(pt_plan,user,&(pt->S0));

                ScanVar(pt_plan,user,&(pt->Divid));

                ScanVar(pt_plan,user,&(pt->R));
                ScanVar(pt_plan,user,&(pt->Sigma0));
                ScanVar(pt_plan,user,&(pt->MeanRevers
ion));
                ScanVar(pt_plan,user,&(pt->LongRunVar
iance));
                ScanVar(pt_plan,user,&(pt->Sigma));
                ScanVar(pt_plan,user,&(pt->Rho));
            }
            while ((model->Show)(user,pt_plan,model)
);

    return ((model->Show)(TOSCREENANDFILE,pt_plan,
        model));
}

```

```

int MOD(Show)(int user,Planning *pt_plan,Model *
    model)
{
    TYPEMOD* pt=(TYPEMOD*)(model->TypeModel);
    VAR r_bs,divid_bs;

    Fprintf(user,"##Model:%s{n",model->Name);

    PrintVar(pt_plan,user,&(pt->T));
    PrintVar(pt_plan,user,&(pt->S0));

    PrintVar(pt_plan,user,&(pt->Divid));
    divid_bs.Val.V_DOUBLE=log(1+pt->Divid.Val.V_
        DOUBLE/100);
    divid_bs.Vtype=DOUBLE;
    strcpy(divid_bs.Vname,"-->Instantaneous Divid
        end Rate");
    divid_bs.Viter=FORBID;
        PrintVar(pt_plan,user,&divid_bs);

    PrintVar(pt_plan,user,&(pt->R));
    r_bs.Vtype=DOUBLE;
    r_bs.Val.V_DOUBLE=log(1+pt->R.Val.V_DOUBLE/100
        );
    r_bs.Vtype=DOUBLE;
    strcpy(r_bs.Vname,"-->Instantaneous Interest
        Rate");
    r_bs.Viter=FORBID;
        PrintVar(pt_plan,user,&r_bs);
    PrintVar(pt_plan,user,&(pt->Sigma0));
    PrintVar(pt_plan,user,&(pt->MeanReversion));

    PrintVar(pt_plan,user,&(pt->LongRunVariance));
    PrintVar(pt_plan,user,&(pt->Sigma));
    PrintVar(pt_plan,user,&(pt->Rho));
    return (model->Check)(user,pt_plan,model);
}

int MOD(Check)(int user,Planning *pt_plan,Model *
    model)
{

```

```

TYPEMOD* pt=(TYPEMOD*)(model->TypeModel);
int status=OK;
char helpfile[MAX_PATH_LEN]="";

if ((2*strlen(model->ID)+strlen("{mod{" +
    strlen("{{" +
    +strlen("_doc.pdf"))>=MAX_PATH_LEN)
{
    Fprintf(TOSCREEN,"%s\n",error_msg[PATH_TOO_
    LONG]);
    exit(WRONG);
}

strcpy(helpfile,path_sep);
strcat(helpfile,"mod");
strcat(helpfile,path_sep);

strcat(helpfile,model->ID);
strcat(helpfile,path_sep);

strcat(helpfile,model->ID);
strcat(helpfile,"_doc.pdf");

status+=ChkVar(pt_plan,&(pt->T));
status+=ChkVar(pt_plan,&(pt->S0));
status+=ChkVar(pt_plan,&(pt->Divid));

status+=ChkVar(pt_plan,&(pt->R));
status+=ChkVar(pt_plan,&(pt->Sigma0));
status+=ChkVar(pt_plan,&(pt->MeanReversion));
status+=ChkVar(pt_plan,&(pt->LongRunVariance))
;
status+=ChkVar(pt_plan,&(pt->Sigma));
status+=ChkVar(pt_plan,&(pt->Rho));

return Valid(user,status,helpfile);
}

int MOD(Init)(Model *model)
{

```

```
TYPEMOD* pt=(TYPEMOD*)(model->TypeModel);
static int first=1;

if (first)
{
    strcpy(pt->T.Vname,"Current Date");
    pt->T.Vtype=DATE;
    pt->T.Val.V_DATE=0.;
    pt->T.Viter=ALLOW;

    strcpy(pt->S0.Vname,"Spot");
    pt->S0.Vtype=PDOUBLE;
    pt->S0.Val.V_PDOUBLE=100.;
    pt->S0.Viter=ALLOW;

    strcpy(pt->Divid.Vname,"Annual Dividend Rate");
    pt->Divid.Vtype=DOUBLE;
    pt->Divid.Val.V_DOUBLE=0.;
    pt->Divid.Viter=ALLOW;

    strcpy(pt->R.Vname,"Annual Interest Rate");
    pt->R.Vtype=DOUBLE;
    pt->R.Val.V_DOUBLE=10.;
    pt->R.Viter=ALLOW;

    strcpy(pt->Sigma0.Vname,"Current Variance")
    ;
    pt->Sigma0.Vtype=DOUBLE;
    pt->Sigma0.Val.V_DOUBLE=0.01;
    pt->Sigma0.Viter=ALLOW;

    strcpy(pt->MeanReversion.Vname,"Mean Reversion");
    pt->MeanReversion.Vtype=DOUBLE;
    pt->MeanReversion.Val.V_DOUBLE=2.;
    pt->MeanReversion.Viter=ALLOW;

    strcpy(pt->LongRunVariance.Vname,"Long-Run Variance");
    pt->LongRunVariance.Vtype=DOUBLE;
```

```
pt->LongRunVariance.Val.V_DOUBLE=0.01;
pt->LongRunVariance.Viter=ALLOW;

strcpy(pt->Sigma.Vname,"Volatility of Volatility");
pt->Sigma.Vtype=DOUBLE;
pt->Sigma.Val.V_DOUBLE=0.2;
pt->Sigma.Viter=ALLOW;

strcpy(pt->Rho.Vname,"Rho");
pt->Rho.Vtype=DOUBLE;
pt->Rho.Val.V_DOUBLE=0.5;
pt->Rho.Viter=ALLOW;

first=0;
}

return OK;
}
```

```
TYPEMOD Heston1dim;

MAKEMOD(Heston1dim);
```

References