Help

```
#include "
href../../mod/hes1d_multifactor/hes1d_multifactor_h_src.pdfhes1d_multifactor.h"
#include "
href../../common/chk_h_src.pdfchk.h"
#include "
href../../common/error_msg_h_src.pdferror_msg.h"
#include "
href../../mod/hes1d/hes1d_pad/model_h_src.pdfmodel.h"

/**
 * Dynamically adjust the of PnlMat stored in a VAR.
 *
 * @param x a VAR
 * @param size the new size
 * @param default_val the diagonal value, all others are zero
 *
 * @return
 */
static int adjust_matrix_size(VAR *x, int size, double default_val)
{
  PnlMat *M = x->Val.V_PNLMAT;

  if (M == NULL)
    {
      if ((x->Val.V_PNLMAT = pnl_mat_create_from_double(size, size, 0.)) == NULL
        return MEMORY_ALLOCATION_FAILURE;
      M = x->Val.V_PNLMAT;
      pnl_mat_set_diag(M, default_val, 0);
      return OK;
    }
  /* If the shape of M is correct, keep the values inside. */
  if ((M->m == M->n) && (M->m == size)) return OK;
  pnl_mat_resize(M, size, size);
  pnl_mat_set_zero(M);
  pnl_mat_set_diag(M, default_val, 0);
  return OK;
}

static void set_vol_size(void *model)
```

```
{
  TYPEMOD *pt = (TYPEMOD *)(model);

  int sz = pt->size.Val.V_PINT;

  adjust_matrix_size(&pt->Sigma0, sz, 0.01);
  adjust_matrix_size(&pt->Correl, sz, -0.7);
  adjust_matrix_size(&pt->Q, sz, 0.25);
  adjust_matrix_size(&pt->M, sz, -3.);

  if (pt->beta.Val.V_DOUBLE < sz - 1)
    {
      pt->beta.Val.V_DOUBLE = sz + 1;
    }
}


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

  if (model->init == 0)
    {
      model->init = 1;
      model->nvar = 0;

      pt->size.Vname = "Volatility size";
      pt->size.Vtype = PINT;
      pt->size.Val.V_PINT = 2;
      pt->size.Viter = ALLOW;
      pt->size.setter = set_vol_size;
      model->nvar++;

      pt->T.Vname = "Current Date";
      pt->T.Vtype = DATE;
      pt->T.Val.V_DATE = 0.;
      pt->T.Viter = FORBID;
      model->nvar++;

      pt->S0.Vname = "Spot";
      pt->S0.Vtype = PDOUBLE;
```

```
pt->S0.Val.V_PDOUBLE = 100.;
pt->S0.Viter = ALLOW;
model->nvar++;

pt->Divid.Vname = "Annual Dividend Rate";
pt->Divid.Vtype = DOUBLE;
pt->Divid.Val.V_DOUBLE = 0.;
pt->Divid.Viter = ALLOW;
model->nvar++;

pt->R.Vname = "Annual Interest Rate";
pt->R.Vtype = DOUBLE;
pt->R.Val.V_DOUBLE = 10.;
pt->R.Viter = ALLOW;
model->nvar++;

pt->Sigma0.Vname = "Initial Volatility";
pt->Sigma0.Vtype = PNLMAT;
pt->Sigma0.Val.V_PNLMAT = NULL;
pt->Sigma0.Viter = FORBID;
model->nvar++;

pt->Correl.Vname = "Correlation";
pt->Correl.Vtype = PNLMAT;
pt->Correl.Val.V_PNLMAT = NULL;
pt->Correl.Viter = FORBID;
model->nvar++;

pt->beta.Vname = "Mean reversion";
pt->beta.Vtype = DOUBLE;
pt->beta.Val.V_DOUBLE = pt->size.Val.V_PINT + 1.;
pt->beta.Viter = FORBID;
model->nvar++;


pt->Q.Vname = "Volatility of variance";
pt->Q.Vtype = PNLMAT;
pt->Q.Val.V_PNLMAT = NULL;
pt->Q.Viter = FORBID;
model->nvar++;
```

```
      pt->M.Vname = "Long run variance";
      pt->M.Vtype = PNLMAT;
      pt->M.Val.V_PNLMAT = NULL;
      pt->M.Viter = FORBID;
      model->nvar++;
    }

  {
    int sz = pt->size.Val.V_PINT;

    adjust_matrix_size(&pt->Sigma0, sz, 0.01);
    adjust_matrix_size(&pt->Correl, sz, -0.7);
    adjust_matrix_size(&pt->Q, sz, 0.25);
    adjust_matrix_size(&pt->M, sz, -3.);
  }

  return OK;
}

TYPEMOD hes1d_multifactor;
MAKEMOD(hes1d_multifactor);
```