Help

```
#include "schwartztrolle.h"
#include "chk.h"
#include "error_msg.h"
#include "model.h"
#include "pnl/pnl_vector.h"

extern char *path_sep;

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

  static double v0[] = {0.1, 0.1};
  static double eta[] = {1.2, 1};
  static double kappa[] = {6.7965, 0.8960, -0.0522, -5.4153};
  static double sigma[] = {0.2085, 0.0704, 8.7350, 1.9223};
  static double rho[] = { -0.8914, -0.0390, -0.1031, -0.1270, -0.1312, -0.000};
  static double alpha[] = {0.3156, 0.0518};
  static double gammac[] = {1.6446, 0.5473};

  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->P0T0.Vname = "Zero-Coupon Bond value with maturity T0";
      pt->P0T0.Vtype = PDOUBLE;
      pt->P0T0.Val.V_PDOUBLE = 0.95;
      pt->P0T0.Viter = ALLOW;
      model->nvar++;

      pt->f0T.Vname = "Initial Futures";
      pt->f0T.Vtype = PDOUBLE;
```

```
pt->f0T.Val.V_PDOUBLE = 98.;
pt->f0T.Viter = ALLOW;
model->nvar++;

pt->v0.Vname = "v01 v02";
pt->v0.Vtype = PNLVECT;
pt->v0.Val.V_PNLVECT = pnl_vect_create_from_ptr(2, v0);
pt->v0.Viter = FORBID;
model->nvar++;

pt->eta.Vname = "eta1 eta2";
pt->eta.Vtype = PNLVECT;
pt->eta.Val.V_PNLVECT = pnl_vect_create_from_ptr(2, eta);
pt->eta.Viter = FORBID;
model->nvar++;

pt->kappa.Vname = "k1 k2 k3 k4";
pt->kappa.Vtype = PNLVECT;
pt->kappa.Val.V_PNLVECT = pnl_vect_create_from_ptr(4, kappa);
pt->kappa.Viter = FORBID;
model->nvar++;

pt->sigma.Vname = "sigma_s1 sigma_s2 sigma_v1 sigma_v2";
pt->sigma.Vtype = PNLVECT;
pt->sigma.Val.V_PNLVECT = pnl_vect_create_from_ptr(4, sigma);
pt->sigma.Viter = FORBID;
model->nvar++;

pt->rho.Vname = "rho_13 rho_15 rho_35 rho_24 rho_26 rho_46";
pt->rho.Vtype = PNLVECT;
pt->rho.Val.V_PNLVECT = pnl_vect_create_from_ptr(6, rho);
pt->rho.Viter = FORBID;
model->nvar++;

pt->alpha.Vname = "alpha1 alpha2";
pt->alpha.Vtype = PNLVECT;
pt->alpha.Val.V_PNLVECT = pnl_vect_create_from_ptr(2, alpha);
pt->alpha.Viter = FORBID;
model->nvar++;

pt->gammac.Vname = "gammac1 gammac2";
```

```
    pt->gammac.Vtype = PNLVECT;
    pt->gammac.Val.V_PNLVECT = pnl_vect_create_from_ptr(2, gammac);
    pt->gammac.Viter = FORBID;
    model->nvar++;

  }


if (pt->sigma.Val.V_PNLVECT == NULL)
  {
    if ((pt->sigma.Val.V_PNLVECT = pnl_vect_create_from_double(4, 0.2)) == NUL
      goto err;
  }

if (pt->rho.Val.V_PNLVECT == NULL)
  {
    if ((pt->rho.Val.V_PNLVECT = pnl_vect_create_from_double(6, -0.5)) == NULL
      goto err;
  }

if (pt->eta.Val.V_PNLVECT == NULL)
  {
    if ((pt->eta.Val.V_PNLVECT = pnl_vect_create_from_double(2, 1.)) == NULL)
      goto err;
  }
if (pt->v0.Val.V_PNLVECT == NULL)
  {
    if ((pt->v0.Val.V_PNLVECT = pnl_vect_create_from_double(2, 0.1)) == NULL)
      goto err;
  }
if (pt->kappa.Val.V_PNLVECT == NULL)
  {
    if ((pt->kappa.Val.V_PNLVECT = pnl_vect_create_from_double(4, 1.)) == NULL
      goto err;
  }
if (pt->alpha.Val.V_PNLVECT == NULL)
  {
    if ((pt->alpha.Val.V_PNLVECT = pnl_vect_create_from_double(2, 0.15)) == NU
      goto err;
  }
```

```
  if (pt->gammac.Val.V_PNLVECT == NULL)
    {
      if ((pt->gammac.Val.V_PNLVECT = pnl_vect_create_from_double(2, 1.0)) == NU
        goto err;
    }


  return OK;

err:
  Fprintf(TOSCREEN, "%s\ n", error_msg[MEMORY_ALLOCATION_FAILURE]);
  exit(WRONG);
}
TYPEMOD Schwartztrolle;
MAKEMOD(Schwartztrolle);
```