

[Help](#)

```
#include "bsdisdiv1d.h"
#include "chk.h"
#include "error_msg.h"
#include "model.h"
#include "pnl/pnl_matrix.h"

extern char *path_sep;

static int adjust_vector_size(VAR *x, int size, double default_value)
{
    PnlVect *v = x->Val.V_PNLVECT;

    if (v == NULL)
    {
        if ((x->Val.V_PNLVECT =
            pnl_vect_create_from_double(size, default_value)) == NULL)
            return MEMORY_ALLOCATION_FAILURE;
        else
            return OK;
    }

    if (v->size == size) return OK;
    return pnl_vect_resize_from_double(v, size, default_value);
}

static void set_Nb_Divid(void *model)
{
    TYPEMOD *pt = (TYPEMOD *) (model);

    adjust_vector_size(&(pt->Amounts), pt->Size.Val.V_PINT, 3.);
    adjust_vector_size(&(pt->Dates), pt->Size.Val.V_PINT, 0.5);
}

static int MOD(Init)(Model *model)
{
    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.;
    pt->T.Viter = ALLOW;
    model->nvar++;

    pt->Size.Vname = "Number of Discrete Dividends";
    pt->Size.Vtype = PINT;
    pt->Size.Val.V_PINT = 1;
    pt->Size.setter = set_Nb_Divid;
    pt->Size.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->Mu.Vname = "Trend";
    pt->Mu.Vtype = DOUBLE;
    pt->Mu.Val.V_DOUBLE = 0.;
    pt->Mu.Viter = ALLOW;
    model->nvar++;

    pt->Sigma.Vname = "Volatility";
    pt->Sigma.Vtype = PDOUBLE;
    pt->Sigma.Val.V_PDOUBLE = 0.2;
    pt->Sigma.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->Amounts.Vname = "Dividend Amounts";
    pt->Amounts.Vtype = PNLVECT;
    pt->Amounts.Val.V_PNLVECT = NULL;
    pt->Amounts.Viter = FORBID;
    model->nvar++;

    pt->Dates.Vname = "Dividend Dates";
    pt->Dates.Vtype = PNLVECT;
    pt->Dates.Val.V_PNLVECT = NULL;
    pt->Dates.Viter = FORBID;
    model->nvar++;
}

adjust_vector_size(&(pt->Amounts), pt->Size.Val.V_PINT, 3.);
adjust_vector_size(&(pt->Dates), pt->Size.Val.V_PINT, 0.5);
return OK;
}

TYPEMOD BlackScholesDisDiv1dim;
MAKEMOD(BlackScholesDisDiv1dim);
```