Help

```
#ifndef GRIDSPARSE_FUNCTIONS_H
#define GRIDSPARSE_FUNCTIONS_H

#include "
href../../../common/math/equity_pricer/gridsparse_constructor_h_src.pdfgridspars

extern int log2int(int x);


extern double GridSparse_real_value_at_points(GridSparse *G, int d, int i);
extern void Nodal_to_Hier_in_dir(int Dir, const PnlVect  *V, PnlVect *Vout, cons
extern void Hier_to_Nodal_in_dir(int Dir, const PnlVect   *V, PnlVect *Vout,  co
extern void Nodal_to_Hier(PnlVect *V, const GridSparse *G);
extern void Hier_to_Nodal(PnlVect *V, const GridSparse *G);
extern PnlVect *V_Hier_to_Nodal(const PnlVect *Vin, const GridSparse *G);
extern double FD_Lap_Stencil_Center(const int i, const int dir,
                                    const PnlVect *v,
                                    GridSparse *G,
                                    int l_Ind_neig,
                                    int r_Ind_neig);
extern double FD_Conv_Stencil_Center(const int i, const int dir,
                                     const PnlVect *v,
                                     const GridSparse *G,
                                     int l_Ind_neig,
                                     int r_Ind_neig);


extern double FD_Conv_Stencil_DeCenter(const int i, const int dir,
                                       const PnlVect *v,
                                       const GridSparse *G,
                                       int Ind_neig);
extern double FD_Lap_Center(const int i, const int dir,
                            const PnlVect *v,
                            GridSparse *G);
extern double FD_Conv_Center(const int i, const int dir,
                             const PnlVect *v,
                             const GridSparse *G);

extern double FD_Conv_DeCenter(const int i, const int dir,
```

```
                                const PnlVect *v,
                                const GridSparse *G,
                                const double coeff);

extern void GridSparse_apply_function(GridSparse *G, PnlVect *Vout, double (*app

extern void GridSparse_fprint(FILE *fic, GridSparse *G,
                              PnlVect *Vout);


extern void GridSparse_Solve_Operator(GridSparse *G, const PnlVect *Vin, PnlVect
                                      void (*operator)(GridSparse *G0,
                                          const PnlVect *V0,
                                          const double a , const double b,
                                          PnlVect *V1),
                                      void (*operator_PC)(GridSparse *G0,
                                          const PnlVect *V0,
                                          const double a, const double b,
                                          PnlVect *V1)
                                     );

typedef struct LaplacienSparseOp
{
  GridSparse *G;
  PnlVect   *V_tmp0;
  PnlVect   *V_tmp1;
} LaplacienSparseOp;


extern LaplacienSparseOp *create_laplacien_sparse_operator();
extern void initialise_laplacien_sparse_operator(LaplacienSparseOp *Op);
extern void laplacien_sparse_operator_free(LaplacienSparseOp **Op);

extern void GridSparse_apply_Laplacien(LaplacienSparseOp *Op, const PnlVect *Vin
                                       const double a , const double b,
                                       PnlVect *Vout);
extern void GridSparse_Solve_Laplacien(LaplacienSparseOp *Op, const PnlVect *Vin
                                       PnlVect *Vout);

typedef struct HeatSparseOp
{
```

```
  double eta;
  double theta_time_scheme;
  PremiaPDETimeGrid *TG;
  GridSparse *G;
  PnlVect  *PC, * V_tmp0, * V_tmp1;
} HeatSparseOp;

extern HeatSparseOp *create_heat_sparse_operator(double eta);
extern void initialise_heat_sparse_operator(HeatSparseOp *Op);
extern void heat_sparse_operator_free(HeatSparseOp **Op);

extern void GridSparse_apply_heat(HeatSparseOp *Op, const PnlVect *Vin,
                                  const double a , const double b,
                                  PnlVect *Vout);
extern void GridSparse_Solve_heat(HeatSparseOp *Op, const PnlVect *Vin,
                                  PnlVect *Vout);

#endif
```