Help

```
#ifndef PDE_TOOLS_H
#define PDE_TOOLS_H

#include "pnl/pnl_vector.h"
#include "pnl/pnl_matrix.h"

/**
 * {defgroup PremiaPDEBoundary  to translate domaine from X
    0,X1 to [0,1]
 */
/*@{*/
typedef struct {
    double X0; /*!< left point */
    double H; /* !< Step */
}PremiaPDEBoundary;

extern PremiaPDEBoundary premia_pde_boundary_create(double
    X0,double X1);
extern double premia_pde_boundary_real_variable(const Prem
    iaPDEBoundary BP ,double X);
extern double premia_pde_boundary_Unit_interval(const Prem
    iaPDEBoundary BP ,double X);
/*@}*/


/**
 * {defgroup PremiaPDEDimBoundary Vector On boundary
 */
/*@{*/
typedef struct PremiaPDEDimBoundary{
  PremiaPDEBoundary * array;
/*!< pointer to store the data */
} PremiaPDEDimBoundary;

extern PremiaPDEDimBoundary* premia_pde_dim_boundary_crea
    te_from_int(int dim);

extern PremiaPDEDimBoundary*
premia_pde_dim_boundary_create(const PnlVect * X0,
             const PnlVect * X1);
```

```
extern void premia_pde_dim_boundary_free(PremiaPDEDimBound
    ary **v);

extern double
premia_pde_dim_boundary_eval_from_unit(double(*f)(const Pn
    lVect* ),
               const PremiaPDEDimBoundary * BP,
               const PnlVect * X);
extern void
premia_pde_dim_boundary_from_unit_to_real_variable(const
    PremiaPDEDimBoundary * BP,
               PnlVect * X);

extern double
premia_pde_dim_boundary_get_step(const PremiaPDEDimBoundar
    y * BP,
         int i);
/*@}*/

typedef struct {
  double current_step;
  int current_index;
  PnlVect * time;
  int is_tuned;
}PremiaPDETimeGrid;

extern PremiaPDETimeGrid * premia_pde_time_homogen_grid(
    const double T,
              const int N_T);
extern void premia_pde_time_grid_free(PremiaPDETimeGrid **
    TG);
extern void premia_pde_time_start(PremiaPDETimeGrid * TG);
extern int premia_pde_time_grid_increase(PremiaPDETimeGrid
    * TG);
extern double premia_pde_time_grid_step(const PremiaPDETime
    Grid * TG);
extern double premia_pde_time_grid_time(const PremiaPDETime
    Grid * TG);


#endif
```

References