

## Help

```
#if defined(PremiaCurrentVersion) && PremiaCurrentVersion < (2008+2) //The "#els
#else
/*****
*   CPS - A simple C PDE solver                               *
*                                                           *
*   Copyright (c) 2007,                                       *
*   Maya Briani      <m.briani@iac.rm.cnr.it>,               *
*   Francesco Ferreri <francesco.ferreri@gmail.com>,         *
*   Roberto Natalini <r.natalini@iac.rm.cnr.it>,             *
*   Marco Papi       <m.papi@iac.rm.cnr.it>                  *
*                                                           *
*****/
#ifndef ASSERTIONS_H
#define ASSERTIONS_H

#include <
href../../common/math/cdo/cdo_math_h_src.pdfmath.h>
#include "
href../../common/math/highdim_solver/laspack/errhandl_h_src.pdfaspack/errhan
#include "
href../../common/math/highdim_solver/cps_debug_h_src.pdfcps_debug.h"

#define OK 0
#define ERR -1

#ifdef ASSERT_ALL
#define ASSERT_PRE 1
#define ASSERT_POST 1
#define ASSERT_CHECK 1
#else
#define ASSERT_PRE 0
#define ASSERT_POST 0
#define ASSERT_CHECK 0
#endif

#define IMPLIES(X,Y) ((!(X)) || (Y))
#define NOT(X) !(X)

typedef enum
```

```

{
    assert_require, assert_ensure, assert_check
} assert_type;

#if ASSERT_PRE != 0
#define REQUIRE(tag,cond) \
if(!(cond)){ \
    assertion_callback(assert_require, __FILE__, __LINE__, __func__, tag, #cond, NULL) \
}
#else
#define REQUIRE(tag,cond)
#endif

#if ASSERT_POST != 0
#define ENSURE(tag,cond) \
if(!(cond)){ \
    assertion_callback(assert_ensure, __FILE__, __LINE__, __func__, tag, #cond, NULL) \
}
#else
#define ENSURE(tag,cond)
#endif

#if ASSERT_CHECK != 0
#define CHECK(tag,cond) \
    if(!(cond)){ \
        assertion_callback(assert_check, __FILE__, __LINE__, __func__, tag, #cond, NULL) \
    }

#define CHECK_LASPACK(tag) \
if (LASResult() != LASOK) { \
WriteLASErrDescr(stderr); \
    assertion_callback(assert_check, __FILE__, __LINE__, __func__, tag, "(LASPACK error)") \
}
#else
#define CHECK(tag,cond)
#define CHECK_LASPACK(tag)
#endif

```

```
#define APPROX_EQUAL(x,y,t) abs((x-y) < t)

void assertion_callback(assert_type, const char *, int, const char *, const char *)
#endif

#endif //PremiaCurrentVersion
```