Help

```
#if defined(PremiaCurrentVersion) && PremiaCurrentVersion < (2008+2) //The "#els
#else
/*****************************************************************************/
/*                                  itersolv.h                               */
/*****************************************************************************/
/*                                                                           */
/* ITERative SOLVers for systems of linear equations                         */
/*                                                                           */
/* Copyright (C) 1992-1995 Tomas Skalicky. All rights reserved.              */
/*                                                                           */
/*****************************************************************************/
/*                                                                           */
/*        ANY USE OF THIS CODE CONSTITUTES ACCEPTANCE OF THE TERMS           */
/*              OF THE COPYRIGHT NOTICE (SEE FILE COPYRGHT.H)                */
/*                                                                           */
/*****************************************************************************/

#ifndef ITERSOLV_H
#define ITERSOLV_H

#include "
href../../../../common/math/highdim_solver/laspack/highdim_vector_h_src.pdfhighd
#include "
href../../../../common/math/highdim_solver/laspack/qmatrix_h_src.pdfqmatrix.h"
#include "
href../../../../common/math/highdim_solver/laspack/precond_h_src.pdfprecond.h"
#include "
href../../../../common/math/highdim_solver/laspack/eigenval_h_src.pdfeigenval.h"
#include "
href../../../../common/math/highdim_solver/laspack/copyrght_h_src.pdfcopyrght.h"

typedef Vector *(*IterProcType)(QMatrix *, Vector *, Vector *, int,
                                PrecondProcType, double);

/* classical iterative methods */

Vector *JacobiIter(QMatrix *A, Vector *x, Vector *b, int NoIter,
                   PrecondProcType Dummy, double Omega);
Vector *SORForwIter(QMatrix *A, Vector *x, Vector *b, int NoIter,
```

```
                    PrecondProcType Dummy, double Omega);
Vector *SORBackwIter(QMatrix *A, Vector *x, Vector *b, int NoIter,
                     PrecondProcType Dummy, double Omega);
Vector *SSORIter(QMatrix *A, Vector *x, Vector *b, int NoIter,
                 PrecondProcType Dummy, double Omega);

/* semi-iterative methods */

Vector *ChebyshevIter(QMatrix *A, Vector *x, Vector *b, int MaxIter,
                      PrecondProcType PrecondProc, double OmegaPrecond);

/* CG and CG-like methods */

Vector *CGIter(QMatrix *A, Vector *x, Vector *b, int MaxIter,
               PrecondProcType PrecondProc, double OmegaPrecond);
Vector *CGNIter(QMatrix *A, Vector *x, Vector *b, int MaxIter,
                PrecondProcType PrecondProc, double OmegaPrecond);
Vector *GMRESIter(QMatrix *A, Vector *x, Vector *b, int MaxIter,
                  PrecondProcType PrecondProc, double OmegaPrecond);
Vector *BiCGIter(QMatrix *A, Vector *x, Vector *b, int MaxIter,
                 PrecondProcType PrecondProc, double OmegaPrecond);
Vector *QMRIter(QMatrix *A, Vector *x, Vector *b, int MaxIter,
                PrecondProcType PrecondProc, double OmegaPrecond);
Vector *CGSIter(QMatrix *A, Vector *x, Vector *b, int MaxIter,
                PrecondProcType PrecondProc, double OmegaPrecond);
Vector *BiCGSTABIter(QMatrix *A, Vector *x, Vector *b, int MaxIter,
                     PrecondProcType PrecondProc, double OmegaPrecond);
void SetGMRESRestart(int MaxSteps);

#endif /* ITERSOLV_H */

#endif //PremiaCurrentVersion
```