

[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 "highdim_vector.h"
#include "qmatrix.h"
#include "precond.h"
#include "eigenval.h"
#include "copyrght.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
```