

Help

```
#if defined(PremiaCurrentVersion) && PremiaCurrentVersion < (2008+2) //The "#els
#else
/*****
/*                                precondition.c                                */
/*****
/*                                */
/* PRECONditioners for iterative solvers of 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)                        */
/*                                */
/*****

#include <
href../../common/math/cdo/cdo_math_h_src.pdfmath.h>

#include "
href../../common/math/highdim_solver/laspac/precond_h_src.pdfaspac/precond
#include "
href../../common/math/highdim_solver/laspac/errhandl_h_src.pdfaspac/errhan
#include "
href../../common/math/highdim_solver/laspac/qmatrix_h_src.pdfaspac/qmatrix
#include "
href../../common/math/highdim_solver/laspac/operats_h_src.pdfaspac/operats
#include "
href../../common/math/highdim_solver/laspac/factor_h_src.pdfaspac/factor.h
#include "
href../../common/math/highdim_solver/laspac/itersolv_h_src.pdfaspac/iterso
#include "
href../../common/math/highdim_solver/laspac/copyrgh_h_src.pdfaspac/copyrg

Vector *JacobiPrecond(QMatrix *A, Vector *y, Vector *c, double Omega)
/* Jacobi preconditioner */
{
    Q_Lock(A);
```

```

V_Lock(y);
V_Lock(c);

/*
 *  Diag(A) / Omega y = c
 *
 *  y = Omega Diag(A)^(-1) c
 */

Asgn_VV(y, Mul_SV(Omega, MulInv_QV(Diag_Q(A), c)));

Q_Unlock(A);
V_Unlock(y);
V_Unlock(c);

return (y);
}

Vector *SSORPrecond(QMatrix *A, Vector *y, Vector *c, double Omega)
/* SSOR preconditioner */
{
    Q_Lock(A);
    V_Lock(y);
    V_Lock(c);

    /*
     *  1 / (2 - Omega) * (Diag(A) / Omega + Lower(A)) * (Diag(A) / Omega)^(-1)
     *      * (Diag(A) / Omega + Upper(A)) y = c
     *
     *  y = (2 - Omega) / Omega * (Diag(A) / Omega + Upper(A))^( -1) * Diag(A)
     *      * (Diag(A) / Omega + Lower(A))^( -1) c
     */

    Asgn_VV(y, Mul_SV((2.0 - Omega) / Omega,
                      MulInv_QV(Add_QQ(Mul_SQ(1.0 / Omega, Diag_Q(A)), Upper_Q(A))
                                Mul_QV(Diag_Q(A),
                                MulInv_QV(Add_QQ(Mul_SQ(1.0 / Omega, Diag_Q(A))
                                Lower_Q(A)),
                                MulInv_QV(Diag_Q(A), c))));

    Q_Unlock(A);
    V_Unlock(y);
    V_Unlock(c);
}

```



```
#endif //PremiaCurrentVersion
```