

NAME

bicg - biconjugate gradient method with preconditioning.

CALLING SEQUENCE

```
[x,err,iter,flag,res] = bicg(A,Ap,b,x0,M,Mp,maxi,tol)
```

PARAMETERS

A : matrix of size n-by-n or function returning $A*x$

Ap : exists if **A** is a function. In this case **Ap** is the transposed function of **A**, i.e returns $A'*x$

b : right hand side vector

x0 : initial guess vector (default: zeros(n,1))

M : preconditioner: matrix or function returning $M*x$ (In the first case, default: eye(n,n))

Mp : must only be provided when **M** is a function. In this case **Mp** is the function which returns $M'*x$

maxi : maximum number of iterations (default: n)

tol : error tolerance (default: 1000*%eps)

x : solution vector

err : final residual norm

iter : number of iterations performed

flag : 0 = **bicg** converged to the desired tolerance within **maxi** iterations
 1 = no convergence given **maxi**
 -1 = breakdown

res : residual vector

DESCRIPTION

Solves the linear system $Ax=b$ using the BiConjugate Gradient Method with preconditioning.

EXAMPLE

```
A=makefish(4); b=rand(16,1);x0=zeros(16,1);
[x,err,iter,flag,res] = bicg(A,b,x0)
M=eye(16,16); max_it=16; tol=1000*%eps;
[x,err,iter,flag,res] = bicg(A,b,x0,M,max_it,tol)

deff("y=precond(x)","y=(M+eye(size(M,1),size(M,2)))*x");
deff("y=precondp(x)","y=(M+eye(size(M,1),size(M,2)))'*x");

deff("y=matvec(x)","y=(A+eye(size(A,1),size(A,1)))*x");
deff("y=matvecp(x)","y=(A+eye(size(A,1),size(A,1)))'*x");

[x,err,iter,flag,res] = bicg(matvec,matvecp,b,x0,precond,precondp,max_it,tol)

[x,err,iter,flag,res] = bicg(matvec,matvecp,b,x0,M,max_it,tol)
[x,err,iter,flag,res] = bicg(A,b,x0,precond,precondp,max_it,tol)
```

AUTHOR

Adaptation by Aladin Group of the corresponding code of netlib/mltemplatesdev (Univ. of Tennessee and Oak Ridge National Laboratory) - 20 March 2001.

SEE ALSO

bicgstab, cgs, gmres, qmr