Jacobi Test

Testing di Robustezza

Testing sul numero di input

```
A=gallery('poisson',10);
xc=jacobi(A);
```

```
Error using jacobi (line 5)
Inserire come input almeno A e b.
```

Testing sulla matrice

Testo se è una matrice

```
A=rand(10,10,10);
b=[1 2 3]';
xc=jacobi(A,b);
```

```
Error using jacobi (line 9)
Il primo input deve essere una matrice.
```

Testo se è sparsa o vuota

```
A=rand(100,100);
x=ones(100,1);
b=A*x;
xc=jacobi(A,b);
```

```
Error using jacobi (line 11)
La matrice deve essere sparsa e non vuota.
```

```
A=[];
A=sparse(A);
b=[1 2 3]';
xc=jacobi(A,b);
```

```
Error using jacobi (line 11)
La matrice deve essere sparsa e non vuota.
```

Testo se non è quadrata

```
A=sparse(rand(100,200));
x=ones(200,1);
b=A*x;
xc=jacobi(A,b);
```

```
Error using jacobi (line 13)
La matrice deve essere quadrata.
```

Testo se è di grandi dimensioni

```
A=gallery('poisson',5);
x=ones(25,1);
b=A*x;
xc=jacobi(A,b);
```

```
Error using jacobi (line 15)
La matrice deve essere di grandi dimensioni(almeno 100x100)
```

Testo se contiene valori corretti

```
A=gallery('poisson',10);
A(15,15)=NaN;
x=ones(100,1);
b=A*x;
xc=jacobi(A,b);
```

```
Error using jacobi (line 17)
Gli elementi della matrice devono essere numeri reali double finiti.
```

```
A=gallery('poisson',10);
A(25,35)=Inf;
x=ones(100,1);
b=A*x;
xc=jacobi(A,b);
```

```
Error using jacobi (line 17)
Gli elementi della matrice devono essere numeri reali double finiti.
```

Controllo se la diagonale ha elementi nulli

```
A=gallery('poisson',10);
A(25,:)=0;
x=ones(100,1);
b=A*x;
xc=jacobi(A,b);
```

```
Error using jacobi (line 22)
Sono presenti elementi nulli all'interno della diagonale principale.
```

Testing sul vettore

Testo se è un vettore colonna

```
A=gallery('poisson',10);
b=10;
xc=jacobi(A,b);
```

```
Error using jacobi (line 28)
Errore, b deve essere un vettore colonna.
```

```
A=gallery('poisson',10);
x=ones(100,1);
b=(A*x)';
xc=jacobi(A,b);
```

```
Error using jacobi (line 28)
Errore, b deve essere un vettore colonna.
```

Testo se ha le stesse dimensioni della matrice

```
A=gallery('poisson',10);
b=[1:200]';
xc=jacobi(A,b);
```

```
Error using jacobi (line 30)
Il vettore b e la matrice A hanno dimensioni diverse.
```

Testo se contiene valori corretti

```
A=gallery('poisson',10);
b=[1:100]';
b(10)=NaN;
xc=jacobi(A,b);
```

```
Error using jacobi (line 32)
Errore, b deve contenere reali finiti double.
```

```
A=gallery('poisson',10);
b=[1:100]';
b(35)=Inf;
xc=jacobi(A,b);
```

```
Error using jacobi (line 32)
```

```
Errore, b deve contenere reali finiti double.
```

Testing su TOL

Testo se è uno scalare

```
A=gallery('poisson',10);
x=ones(100,1);
TOL=[1:10];
b=A*x;
xc=jacobi(A,b,TOL);
```

```
Error using jacobi (line 40)
Errore, TOL deve essere uno scalare
```

Testo se ha valore accettabile

```
A=gallery('poisson',10);
x=ones(100,1);
TOL=Inf;
b=A*x;
xc=jacobi(A,b,TOL);
```

```
Error using jacobi (line 42)
Errore, TOL deve essere settato come un numero reale.
```

```
A=gallery('poisson',10);
x=ones(100,1);
TOL=NaN;
b=A*x;
xc=jacobi(A,b,TOL);
```

```
Error using jacobi (line 42)
Errore, TOL deve essere settato come un numero reale.
```

Testing su MAXITER

Testo se è uno scalare

```
A=gallery('poisson',10);
x=ones(100,1);
NMAX=[1:10]';
b=A*x;
xc=jacobi(A,b,[],NMAX);
```

```
Error using jacobi (line 50)
Errore, MAXITER deve essere uno scalare
```

Testo se ha valore accettabile

```
A=gallery('poisson',10);
x=ones(100,1);
NMAX=Inf;
b=A*x;
xc=jacobi(A,b,[],NMAX);
```

```
Error using jacobi (line 52)
Errore, MAXITER deve essere settato come un numero reale finito.
```

```
A=gallery('poisson',10);
x=ones(100,1);
NMAX=NaN;
b=A*x;
xc=jacobi(A,b,[],NMAX);
```

```
Error using jacobi (line 52)
Errore, MAXITER deve essere settato come un numero reale finito.
```

Testing di Funzionamento

Testo il funzionamento con due parametri di input

```
A=gallery('poisson',10);
c=condest(A)
```

```
c =
  69.863370896510276
```

```
x=ones(100,1);
b=A*x;
xc=jacobi(A,b)
```

```
xc = 100×1
   0.999999072067297
   0.999998219310184
   0.999997510813969
   0.999997003976802
   0.999996739859617
   0.999996739859617
   0.999997003976802
   0.999997510813969
   0.999998219310184
   0.999999072067297
```

⋮

Testo il funzionamento con tre parametri di input

```
A=gallery('poisson',10);
c=condest(A)
```

```
c =
  69.863370896510276
```

```
x=ones(100,1);
b=A*x;
[xc, niter]=jacobi(A,b,10^-4)
```

```
xc = 100×1
   0.999904756571118
   0.999817229198409
   0.999744508829065
   0.999692486834926
   0.999665377728570
   0.999665377728570
   0.999692486834926
   0.999744508829065
   0.999817229198409
   0.999904756571118
     ⋮
niter =
   174
```

Testo il funzionamento con quattro parametri di input

```
A=gallery('poisson',10);
c=condest(A)
```

```
c =
  69.863370896510276
```

```
x=ones(100,1);
b=A*x;
[xc, niter,resrel]=jacobi(A,b,10^-6,400)
```

```
xc = 100×1
   0.999999072067297
   0.999998219310184
   0.999997510813969
   0.999997003976802
   0.999996739859617
   0.999996739859617
   0.999997003976802
   0.999997510813969
   0.999998219310184
   0.999999072067297
```

```
      ⋮
niter =
   286
resrel =
     1.503747201926131e-06
```

Testo il funzionamento se ometto TOL o MAXITER

```
A=gallery('poisson',10);
c=condest(A)
```

```
c =
  69.863370896510276
```

```
x=ones(100,1);
b=A*x;
[xc, niter]=jacobi(A,b,[],400)
```

```
xc = 100×1
   0.999999072067297
   0.999998219310184
   0.999997510813969
   0.999997003976802
   0.999996739859617
   0.999996739859617
   0.999997003976802
   0.999997510813969
   0.999998219310184
   0.999999072067297
      ⋮
niter =
   286
```

```
A=gallery('poisson',10);
c=condest(A)
```

```
c =
  69.863370896510276
```

```
x=ones(100,1);
b=A*x;
[xc, niter]=jacobi(A,b,10^-4,[])
```

```
xc = 100×1
   0.999904756571118
   0.999817229198409
   0.999744508829065
   0.999692486834926
   0.999665377728570
   0.999665377728570
   0.999692486834926
```

```
   0.999744508829065
   0.999817229198409
   0.999904756571118
      ⋮
niter =
   174
```

Testo il funzionamento con valori di TOL errati

```
A=gallery('poisson',10);
c=condest(A)
```

```
c =
  69.863370896510276
```

```
x=ones(100,1);
b=A*x;
[xc, niter]=jacobi(A,b,realmin)
```

```
Warning: Valore TOL errato. Utilizzo valore di default.
xc = 100×1
   0.999999072067297
   0.999998219310184
   0.999997510813969
   0.999997003976802
   0.999996739859617
   0.999996739859617
   0.999997003976802
   0.999997510813969
   0.999998219310184
   0.999999072067297
      ⋮
niter =
   286
```

```
A=gallery('poisson',10);
c=condest(A)
```

```
c =
  69.863370896510276
```

```
x=ones(100,1);
b=A*x;
[xc, niter]=jacobi(A,b,-10^-8)
```

```
Warning: Valore TOL errato. Utilizzo valore di default.
xc = 100×1
   0.999999072067297
   0.999998219310184
   0.999997510813969
   0.999997003976802
   0.999996739859617
```

```
   0.999996739859617
   0.999997003976802
   0.999997510813969
   0.999998219310184
   0.999999072067297
      ⋮
niter =
   286
```

Testo il funzionamento con valori di MAXITER errati

```
A=gallery('poisson',10);
c=condest(A)
```

```
c =
  69.863370896510276
```

```
x=ones(100,1);
b=A*x;
[xc, niter]=jacobi(A,b,[],-5)
```

```
Warning: Valore MAXITER errato. Utilizzo valore di default.
xc = 100×1
   0.999999072067297
   0.999998219310184
   0.999997510813969
   0.999997003976802
   0.999996739859617
   0.999996739859617
   0.999997003976802
   0.999997510813969
   0.999998219310184
   0.999999072067297
      ⋮
niter =
   286
```

```
A=gallery('poisson',10);
c=condest(A)
```

```
c =
  69.863370896510276
```

```
x=ones(100,1);
b=A*x;
[xc, niter]=jacobi(A,b,[],1)
```

```
Warning: Valore MAXITER errato. Utilizzo valore di default.
xc = 100×1
   0.999999072067297
   0.999998219310184
   0.999997510813969
```

```
   0.999997003976802
   0.999996739859617
   0.999996739859617
   0.999997003976802
   0.999997510813969
   0.999998219310184
   0.999999072067297
      ⋮
niter =
   286
```

Testo il funzionamento quando l'algoritmo si arresta senza convergere

```
A=gallery('poisson',10);
c=condest(A)
```

```
c =
  69.863370896510276
```

```
x=ones(100,1);
b=A*x;
[xc]=jacobi(A,b,[],50)
```

```
Warning: L 'algoritmo si è arrestato all'iterazione 50, senza convergere. Il residuo relativo è 0.026020.
xc = 100×1
   0.983938530159793
   0.969180251057885
   0.956919822702248
   0.948151504962609
   0.943582361252061
   0.943582361252061
   0.948151504962609
   0.956919822702248
   0.969180251057885
   0.983938530159793
      ⋮
```

Testing di Accuratezza

Verifica dell'errore relativo per metodo convergente al variare di TOL

```
A=gallery('poisson',10);
c=condest(A)
```

```
c =
  69.863370896510276
```

```
x=ones(100,1);
b=A*x;
[xc, niter,resrel]=jacobi(A,b);
```

```
err=norm(x-xc)/norm(xc)
```

```
err =
     6.429949363240837e-06
```

```
A=gallery('poisson',10);
c=condest(A)
```

```
c =
  69.863370896510276
```

```
x=ones(100,1);
b=A*x;
[xc, niter,resrel]=jacobi(A,b,10^-4);
err=norm(x-xc)/norm(xc)
```

```
err =
     6.603524321910592e-04
```

```
A=gallery('poisson',10);
c=condest(A)
```

```
c =
  69.863370896510276
```

```
x=ones(100,1);
b=A*x;
[xc, niter,resrel]=jacobi(A,b,10^-8);
err=norm(x-xc)/norm(xc)
```

```
err =
     6.528971632873157e-08
```