Analisi Numerica I
Sistemi lineari: metodi iterativi

Ana Alonso

ana.alonso@unitn.it

8 novembre 2019

Metodi iterativi classici

$$A\mathbf{x} = \mathbf{b} \qquad A = P - N$$

$$P\mathbf{x} = N\mathbf{x} + \mathbf{b}$$

$\mathbf{x}^{(0)}$assegnato

For $k \geq 0$

$$P\mathbf{x}^{(k+1)} = N\mathbf{x}^{(k)} + \mathbf{b}$$

$B = P^{-1}N$ si dice matrice d'iterazione del metodo.

- Jacobi: `P=diag(diag(A))`
- Gauss-Seidel: `P=tril(A)`

Test d'arresto

- Bassato sull'incremento:

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| < \epsilon \rightsquigarrow \text{ STOP.}$$

$\mathbf{x} - \mathbf{x}^{(k)} = \mathbf{x} - \mathbf{x}^{(k+1)} + \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} = B(\mathbf{x} - \mathbf{x}^{(k)}) + \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$.
$\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \|B\| \, \|\mathbf{x} - \mathbf{x}^{(k)}\| + \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|$. Se $\|B\| < 1$ allora
$\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \frac{1}{1-\|B\|} \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|$.

- Bassato sul residuo:

$$\|\mathbf{b} - A\mathbf{x}^{(k)}\| < \epsilon \|\mathbf{b}\| \rightsquigarrow \text{ STOP.}$$

$\mathbf{r}^{(k)} = b - A\mathbf{x}^{(k)} = A(\mathbf{x} - \mathbf{x}^{(k)}) \rightsquigarrow \|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \|A^{-1}\| \, \|\mathbf{r}^{(k)}\|$. Siccome $\|\mathbf{b}\| \leq \|A\| \, \|\mathbf{x}\|$ allora $\frac{1}{\|\mathbf{x}\|} \leq \frac{\|A\|}{\|\mathbf{b}\|}$ quindi $\frac{\|\mathbf{x}-\mathbf{x}^{(k)}\|}{\|\mathbf{x}\|} \leq \|A\| \|A^{-1}\| \frac{\|\mathbf{r}^{(k)}\|}{\|\mathbf{b}\|}$

Il metodo di Jacobi

$$A\mathbf{x} = \mathbf{b} \qquad D\mathbf{x} = \mathbf{b} + (D - A)\mathbf{x}$$

for $i = 1 : n$

$$x_i^{(k+1)} = \frac{1}{a_{i,i}} \left(b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{(k)} - \sum_{j=i+1}^{n} a_{i,j} x_j^{(k)} \right)$$

```
function [x,nit]=jacobi(A,b,x0,toll,nmax)
P=diag(A);
N=diag(P)-A;
for nit=1:nmax;
    x=(N*x0+b)./P;
    if norm(x-x0) < toll, return, end
    x0=x;
end
```

Il test d'arresto usa l'incremento.

Esercizio

Scrivere una funzione di Matlab che implementi il metodo iterativo di Gauss-Seidel per l'approssimazione della soluzione di un sistema lineare $A\mathbf{x} = \mathbf{b}$.

for $i = 1 : n$

$$x_i^{(k+1)} = \frac{1}{a_{i,i}} \left(b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{i,j} x_j^{(k)} \right)$$

Usare il test d'arresto basato sul residuo.

Esercizio

Scrivere una funzione di Matlab che implementi il metodo iterativo di Gauss-Seidel per l'approssimazione della soluzione di un sistema lineare $A\mathbf{x} = \mathbf{b}$.

for $i = 1 : n$

$$x_i^{(k+1)} = \frac{1}{a_{i,i}} \left(b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{i,j} x_j^{(k)} \right)$$

Usare il test d'arresto basato sul residuo.

```
function [x,nit]=GS(A,b)
nmax=100; toll=1.e-8;
toll=toll*norm(b);
x=b;
n=length(b);
for nit=1:nmax;
    for i=1:n
        x(i)=(b(i)-A(i,1:i-1)*x(1:i-1)-A(i,i+1:n)*x(i+1:n))/A(i,i);
    end
    if norm(b-A*x) < toll, return, end
end
```

Esercizio

Data la matrice

$$\begin{bmatrix} 9 & 14 & -3 \\ 14 & 24 & -6 \\ -3 & -6 & 5 \end{bmatrix}$$

- Verificare che è simmetrica definita positiva.
- Verificare che il metodo di Jacobi non converge.
- Verificare che il metodo di Richardson

  $$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha P^{-1}(\mathbf{b} - A\mathbf{x}^{(k)})$$

  con $P = D$ e $\alpha = 0.85$ converge.

Matrici sparse

Si dice che una matrice è sparsa se ha “tanti” coefficienti uguali a zero. (Una matrice tridiagonale è un esempio di matrice sparsa.)
Se una matrice è sparsa è conveniente memorizare solo dove si trovano gli elementi non nulli e il loro valore.

- Il comando `sparse` memoriza una matrice in questo modo.
- Il comando `nnz` restituisce il numero di elementi non nulli di una matrice.
- Il comando `spy` “disegna” la struttura di una matrice.

I metodi diretti “riempiono” le matrici sparse

Esempi di matrici sparse

- Le matrici tridiagonali.
  - Il comando `spdiags` ci permette di creare matrici a banda in formato `sparse`.
- Le matrici risultanti del metodo degli elementi finiti.
  - Il file `matrix.mat` contiene una di queste matrici.
  - In questo esempio la matrice è simmetrica definita positiva.

Il metodo del gradiente coniugato

$$A\mathbf{x} = \mathbf{b}$$

con A simmetrica definita positiva.

$$\begin{aligned}
&\mathbf{x}^{(0)}\text{assegnato} \\
&\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}\,, \quad \mathbf{d}^{(0)} = \mathbf{r}^{(0)} \\
&\text{for } k \geq 0 \\
&\quad \alpha_k = \frac{(\mathbf{r}^{(k)}, \mathbf{d}^{(k)})}{(A\mathbf{d}^{(k)}, \mathbf{d}^{(k)})} \\
&\quad \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)} \\
&\quad \mathbf{r}^{(k+1)} = \mathbf{b} - A\mathbf{x}^{(k+1)} \\
&\quad \beta_k = -\frac{(A\mathbf{r}^{(k+1)}, \mathbf{d}^{(k)})}{(A\mathbf{d}^{(k)}, \mathbf{d}^{(k)})} \\
&\quad \mathbf{d}^{(k+1)} = \mathbf{r}^{(k+1)} + \beta_k \mathbf{d}^{(k)}
\end{aligned}$$

Esercizio

Scrivere una funzione di Matlab che implementi il metodo del gradiente coniugato. (Usare il test d'arresto basato sul residuo.)

Esercizio

Scrivere una funzione di Matlab che implementi il metodo del gradiente coniugato. (Usare il test d'arresto basato sul residuo.)

```
function [x,k]=gradcon(A,b,toll,nmax,x)
toll=toll*norm(b);
r=b-A*x;
if norm(r) < toll, k=0; return, end
d=r;
for k=1:nmax
  alpha=(d'*r)/(d'*A*d);
  x=x+alpha*d;
  r=b-A*x;
  if norm(r) < toll, return, end
  beta=-(d'*A*r)/(d'*A*d);
  d=r+beta*d;
end
```

Esempio

```
>> load matrix
>> spy(A)
>> b=A*ones(3447,1);
>> [x,M]=gaussNP(A,b);
>> spy(M)
>> AA=sparse(A);
>> [x,M]=gaussNP(AA,b);
>> x=A\b;
>> [x,k]=gradcon(AA,b,1.e-8,500,b);
```