Progetto Calcolo Scientifico: Regolarizzazione di sistemi lineari malcondizionati con la SVD troncata e criterio della curva L

Matteo Scardala
m.scardala@studenti.unipi.it

Marzo 2024

1 Introduzione

Un problema fondamentale dell'Algebra Lineare Numerica è il problema dei minimi quadrati, cioé trovare il vettore $\tilde{x} = arg\min_x \|Ax - b\|_2$
In questo lavoro vengono riportati i risultati di un esperimento che riguarda la risoluzione di un sistema lineare riconducendosi al problema dei minimi quadrati nel caso di una matrice A che ha un elevato numero di condizionamento perché i suoi valori singolari decrescono velocemente a zero (in questo caso si dice che numericamente è di rango non pieno).
Lo scopo è infatti cercare un metodo stabile rispetto a perturbazioni del vettore dei termini noti b.

2 Background teorico e notazione

Sia $Ax = b$, $A \in \mathbf{R}^{m\times n}, n \leq m$. Sia $A = U\Sigma V^T$ la sua SVD e $\{\sigma_i\}_{i=1,...,n}$ i valori singolari in ordine non crescente. Siano $\{u_i\}_{i=1,...,n}$ e $\{v_i\}_{i=1,...,n}$ rispettivamenti i vettori singolari sinistri e destri.
Dalla teoria sappiamo che nel caso di sistemi rettangolari sovradeterminati, sottodeterminati o di rango non pieno possiamo trovare una "buona" soluzione x_0 al probelma riconducendosi a trovare $x_0 = arg\min_x \|Ax-b\|_2$ di minima norma.
Inoltre è noto che la soluzione di minima norma del problema dei minimi quadrati così formulato è $x_0 = A^\dagger b = \sum_{i=0}^{n} \frac{u_i^T b}{\sigma_i} v_i$ dove $A^\dagger = V\Sigma^\dagger U^T$,
$\Sigma^\dagger = diag(\sigma_1^\dagger, ..., \sigma_n^\dagger)$ dove $y^\dagger = y^{-1}$ se $y \neq 0$, $y^\dagger = 0$ altrimenti.
La matrice A con cui vogliamo lavorare ha valori singolari che decrescono velocemente a zero, dunque la divisione per σ_i da un certo punto in poi non può che amplificare notevolmente l'eventuale errore sul vettore dei termini noti e rendere la soluzione insoddisfaccente. Per questo motivo introduciamo due metodi di

regolarizzazione del problema ai minimi quadrati:

1. **Il metodo di Tikhonov:** invece che risolvere $x_0 = arg\min_x \|Ax - b\|_2$, risolviamo $\tilde{x} = arg\min_x \{\|Ax - b\|_2{}^2 + \lambda^2 \|x\|_2^2\}$.
   λ viene chiamato parametro di regolarizzazione e la soluzione corrispondente $x_\lambda = \sum_{i=0}^{n} \frac{\sigma_i}{\sigma_i^2 + \lambda^2} u_i^T b v_i$ è nota come soluzione regolarizzata.

2. **SVD troncata:** invece di risolvere il problema per la matrice A, consideriamo $A_k = U\Sigma_k V^T$, che è la migliore approssimazione di rango k di A, dove Σ_k è la matrice dei valori singolari troncata fino al $k-$esimo valore;
   In questo caso la soluzione risulta essere $x_k = \sum_{i=0}^{k} \frac{u_i^T b}{\sigma_i} v_i$.
   k viene chiamato parametro di troncamento, mentre x_k soluzione TSVD.

Sia δb la perturbazione di b, mentre $\tilde{x}_k = x_k + \delta x_k$ e $\tilde{x}_\lambda = x_\lambda + \delta x_\lambda$ rispettivamente la soluzione troncata e regolarizzata del problema con b perturbato.
Si può mostrare in entrambi i casi che ricondursi alla regolarizzazione rende il problema meglio condizionato:

$$\frac{\|\delta x_k\|}{\|x_k\|} \leq \frac{\sigma_1}{\sigma_k} \frac{\|\delta b\|}{\|Ax_k\|}$$

$$\frac{\|\delta x_\lambda\|}{\|x_\lambda\|} \leq \frac{\sigma_1}{2\lambda} \frac{\|\delta b\|}{\|Ax_\lambda\|}$$

È immediato notare che il condizionamento è migliore tanto più λ è grande e k piccolo.
Adesso studiamo come le soluzioni regolarizzate e troncate vengono influenzate dalle perturbazione del vettore dei termini noti.
Definiamo

$$r_k = b_0 - Ax_k = \sum_{i=k+1}^{n} u_i^T b u_i$$

$$r_\lambda = b_0 - Ax_\lambda = \sum_{i=1}^{n} \frac{\lambda^2}{\sigma_i^2 + \lambda^2} u_i^T b u_i$$

i residui corrispondenti a x_k e a x_λ, dove $b_0 = Ax_0$. Si osserva che questi non sono i residui esatti, come $b - Ax_k$, ma li possiamo vedere come le componenti dei residui che stanno in $colspan(A)$, cioé la parte del residuo che veramente cambia al variare dei parametri e che vogliamo studiare.
Se è vero che al crescere di λ il problema è meglio condizionato, d'altra parte per valori grandi del parametro di regolarizzazione il residuo risulta stabilizzarsi attorno al valore b_0. Viceversa se scegliamo un parametro piccolo otteniamo soluzioni in teoria più soddisfacienti, perché con residuo più piccolo, ma molto pertrubate dagli errori. Adesso introduciamo un criterio per cercare il parametro λ ottimale in questo *trade-off* tra pertrubazione e grado di soddisfazione della soluzione.

Supponiamo che b soddisfi la DPC[1], mentre δb no. In particolare nell'esperimento prenderemo δb tale che tutte le perturbazioni $|u_i^T \delta b| \approx \epsilon_0$ siano simili al variare di i. In un'analisi approssimata possiamo assumere che i termini pertrubati dei residui e delle soluzioni siano quelli che si ottengono ponendo il vettore dei termini noti uguale a δb, quindi abbiamo

$$\delta x_\lambda \approx \epsilon_0 \sum_{i=0}^{n} \frac{\sigma_i}{\sigma_i^2 + \lambda^2} v_i \qquad \delta r_\lambda \approx \epsilon_0 \sum_{i=1}^{n} \frac{\lambda^2}{\sigma_i^2 + \lambda^2} u_i$$

Data la condizione DPC su b, abbiamo che i valori $|u_i^T b|$ vanno a zero molto velocemente (poiché lo fanno i valori singolari di A), mentre $|u_i^T \delta b|$ resta stabile attorno a ϵ_0. Di conseguenza

1. Per valori di $\lambda << \sigma_n$ in $x_\lambda + \delta x_\lambda$ domina il termine di perturbazione.
2. Per valori di $\lambda >> \sigma_n$ in $x_\lambda + \delta x_\lambda$ domina il termine non perturbato.

Questo comportamento genera una curva nel piano $(\|r_\lambda\|, \|x_\lambda\|)$ che forma una "L". Il parametro λ ottimale per ottenere soluzioni poco pertrubate (valori di $\|\tilde{x_\lambda}\|$ vicini a $\|x_0\|$) e con residuo piccolo è quello che si ha in corrispondenza dell'angolo della "L". Questo metodo con cui si trova il parametro di regolarizzazione viene chiamato criterio della curva L.
In realtà il parametro λ ottimale ci permette anche di ottenere il parametro di troncamento ottimale, dato che alcuni risultati teorici ci garantiscono che, sotto l'ipotesi di b con DPC, eiste $\lambda \in [\sigma_{k+1}, \sigma_k]$ tale che x_λ e x_k sono "vicini".

3 Esperimento e risultati

L'esperimento nasce dalla seguente equazione integrale in f

$$\int K(s,x) f(x) dx = g(s)$$

È possibile discretizzare il problema fissando $n \in \mathbf{N}$ e una base ortonormale di funzioni

$$\Phi_i(x) = \begin{cases} h^{-1/2} & \text{se } a + (i-1)h \leq x \leq a + ih \\ 0 & \text{altrimenti} \end{cases} \quad i = 1, ..., n$$

dove $[a, b]$ è l'intervallo di integrazione e $h = \frac{b-a}{n}$.
Le coordinate della soluzione in questa base sono fornite dal vettore che risolve il problema dei minimi quadrati relativo al sistema $Ax = b$ dove

$$A_{ij} = h^{-1} \int_{a+(i-1)h}^{a+ih} \int_{a+(j-1)h}^{a+jh} K(s,x) dx ds$$

[1]DPC (Discrete Picard Condition): stiamo assumendo che per gli indici i corrispondenti a valori singolari non nulli i valori di $|u_i^T b|$ decadono a zero più velocemente dei valori singolari σ_i

$$b_i = h^{-1/2} \int_{a+(i-1)h}^{a+ih} g(s)ds$$

Nell'esperimento abbiamo posto

$$K(s,x) = \begin{cases} 1 + cos[\pi(s-x)/3] & \text{se } |s-x| \leq 3 \\ 0 & \text{se } |s-x| > 3 \end{cases}$$

$$g(s) = (6 - |s|)(1 + 1/2\cos(\frac{\pi s}{3})) + \frac{9}{2\pi}\sin(\frac{\pi|s|}{3})$$

$[a,b] = [-6,6]$ e $n = 64$. Inoltre abbiamo perturbato il vettore b con un vettore casuale le cui componenti sono state generate indipendentemente da una distribuzione gaussiana con media nulla e varianza $\epsilon_0 = 10^{-4}$, dunque $\|\delta b\| \approx 8 \cdot 10^4$. Nel codice 1 ho costruito la matrice A e il vettore b pertrubato. Inoltre tramite la function `[norma_x,norma_r]= norme_lambda(U,s,V,b_vect,n,lambda)` calcolo $\|\tilde{x}_\lambda\|$ e $\|\tilde{r}_\lambda\|$ (dove $\tilde{r}_\lambda = r_\lambda + \delta r_\lambda$). Questi valori vengono riportati su un grafico (con scala logartimica sull'asse delle ascisse) come spiegato nella sezione precedente (Figura 1).
Inoltre su un altro grafico (Figura 2) con scala logaritmica sull'asse delle ordinate ho riportato i valori σ_i (puntini rossi), $|u_i^T \tilde{b}|$ (pallini blu) e $|u_i^T \tilde{b}/\sigma_i|$ (crocette nere).

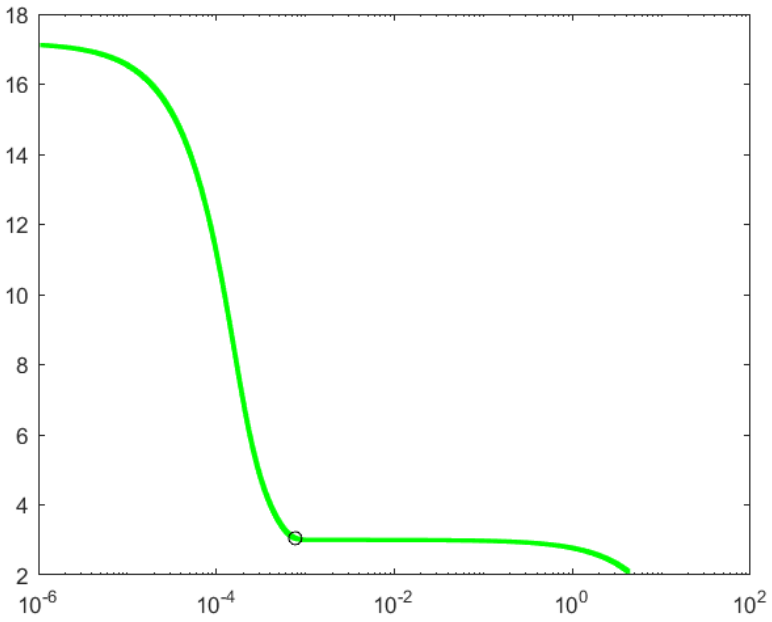

Figura 1: Curva a "L". $\|\tilde{x}_\lambda\|$ sull'asse verticale e $\|\tilde{r}_\lambda\|$ sull'asse orizzontale. Il cerchietto nero evidenzia il livello $\lambda = 6.8 \cdot 10^{-4}$

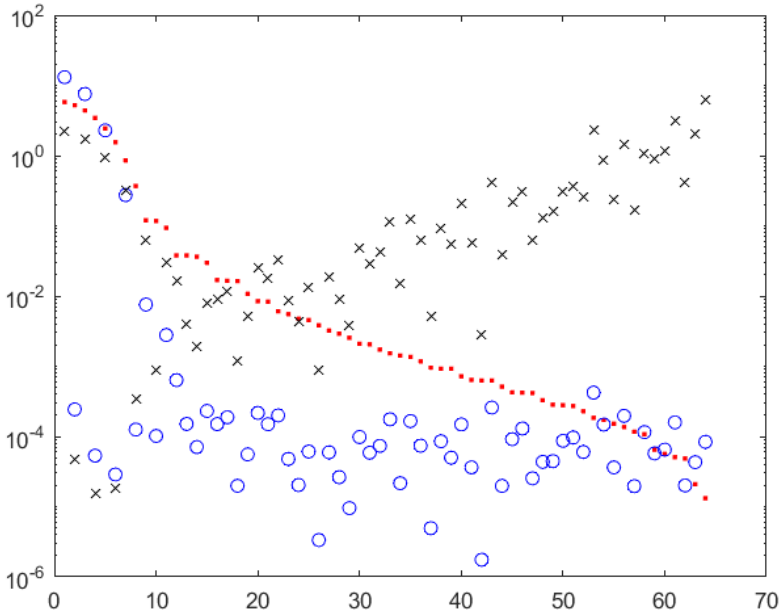

Figura 2: Con scala logaritmica sull'asse delle ordinate ho riportato i valori σ_i (puntini rossi), $|u_i^T \tilde{b}|$ (pallini blu) e $|u_i^T \tilde{b}/\sigma_i|$ (crocette nere).

Osservazioni: coerentemente a quanto spiegato nella sezione precedente il grafico in figura 1 ha una forma a L. Con un cerchietto nero ho evidenziato l'angolo della "L" che corrisponde al valore di lambda ottimale ($\lambda = 6.8 \cdot 10^{-4}$). Come ho scelto questo valore: l'angolo della curva si trova nel punto$(7.69284 \cdot 10^{-4}, 3.049)$. Dunque ho considerato il vettore `w=(x<=3.050).*(x>=3.048)`. Nel codice il vettore x contiene le norme $\|\tilde{x}_\lambda\|$ al variare di λ. Tramite il comando `find(w)` ho scoperto dunque che il valore di $\|\tilde{x}_\lambda\|$ corrispondente al punto angoloso della "L" si trova tra l'indice 4353 e 4365 del vettore x. Per come ho definito `lambda` nel codice, vuol dire che il valore di λ cercato si trova nell'intervallo $[10^{-6} + 4353 * 6.5 * 10^{-4}, 10^{-6} + 4365 * 6.5 * 10^{-4}] \approx [6.75, 6.87] \cdot 10^{-4}$ dunque $\lambda \approx 6.8 \cdot 10^{-4}$
Nel secondo grafico si nota che effettivamente i valori di $|u_i^T \tilde{b}|$ si stabilizzano attorno al valore 10^{-4}. Infatti il vettore b soddisfa la DPC e i valori $|u_i^T b|$ decadono a zero velocemente, mentre il contributo $|u_i^T \delta b| \approx 10^{-4}$ resta costante e dunque diventa dominante. Quando il contributo della perturbazione diventa dominante il rapporto $|u_i^T b/\sigma_i|$ (crocette nere) inizia a crescere perché diventa un rapporto tra un termine quasi costante e un termine che decresce. Questo comportamento del grafico ci indica che il valore di troncamento k ottimale si trova in corrispondenza della regione in cui inizia a diventare dominante il termine perturbato e cioè dove il grafico delle crocette nere smette di decrescere

e inizia a crescere (creando una sorta di minimo locale). Nel nostro caso particolare questa regione si trova tra i valori degli indici tra 12 e 22. Questo è vero perché la forma a "L" del grafico si forma quando il termine pertrubato inizia a diventare dominante. I grafici 1 e 2 sono stati ottenuti grazie ai codici 1 e 2.

In seguito ho fissato il valore di $\lambda = 6.8 \cdot 10^{-4}$ che si è dimostrato ottimale nel caso precedente. In corrispondenza di questo valore ho ripetuto l'esperimento per valori di n, posti a 64,128,256,512,1024. Questa volta ho studiato i tempi computazionali. Nel grafico in Figura 3 in scala logaritmica sono confrontati i tempi di calcolo e la taglia della matrice. I punti sembrano allineati. Un comportamento del genere in in un grafico loglog è tipico dei monomi x^n o in generale dei polinomi nelle zone in cui un termine domina. Questo risultato è coerente con la teoria: l'algortimo consiste in una SVD a cui seguono delle operazioni aritmetiche, il costo è dunque $O(n^3)$. Infatti le operazioni dentro il ciclo `for` costano $O(n)$, quindi le operazioni aritmetiche dopo la SVD sono $O(n^2)$. Il costo di `svd` dovrebbe essere $O(n^3)$ in quanto si riconduce al calcolo di autovalori/autovettori di una matrice che al massimo ha taglia doppia. Il costo dominante è dunque $O(n^3)$ cioé polinomiale di grado 3.
Infine ho valutato il comportamento dei valori singolari al crescere di n. Intanto valutando a ogni iterazione s(1) si scopre che in tutti i casi il valore singolare maggiore cambia solo leggermente (meno 1 percento). Nella figura 4 è riportato il numero di valori singolari maggiori di 10^{-6}. Sembrerebbe che questo numero cresca in modo monotono ma crescendo sempre di meno, un andamento simile a un logaritmo. Questo fa presumere che al crescere di n la matrice sia sempre più mal condizionata e numericamente di rango sempre molto più basso rispetto alla taglia della matrice. Questo comportamento è confermato dal grafico in figura 5 che riporta la percentuale di valori singolari maggiori di 10^{-6} che decresce (sempre di più). Lo studio delle proprietà al variare di n è stato fatto con il codice 3.

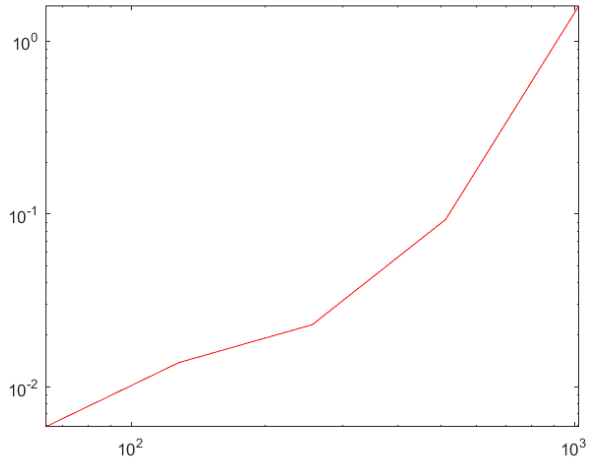

Figura 3: Grafico loglog che confronta il tempo di calcolo per la soluzione del sistema con la taglia della matrice. Il grafico sembra una retta: coerente con un costo polinomiale di grado 3 con n grande.

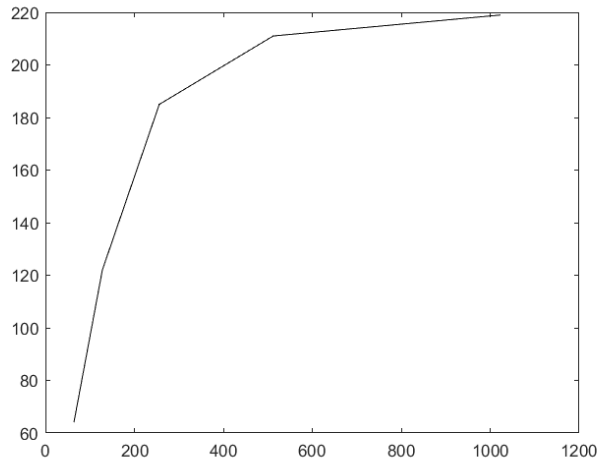

Figura 4: Nel grafico sono confrontati il numero di valori singolari maggiori di una soglia $\epsilon = 10^{-6}$ con la taglia della matrice

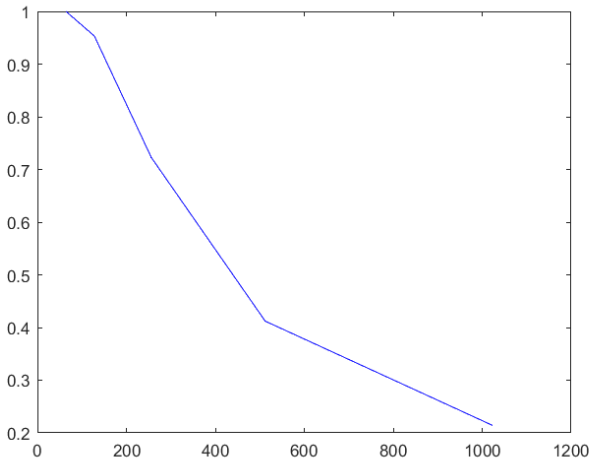

Figura 5: Nel grafico sono confrontati la percentuale dei valori singolari maggiori di una soglia $\epsilon = 10^{-6}$ con la taglia della matrice

4 Codici

Codice 1

```
%Per prima cosa genero la matrice A e il vettore dei termini noti b
pi=3.14159265358979323846264338327950288419716939937510582097494459230781640628;
n=64;
a=-6;
b=6;
h=(b-a)/n;

K= @(x,s) (1+cos(pi*(s-x)/3)).*(abs(s-x)<=3);
g= @(s) (6-abs(s)).*(1+0.5*cos(pi*s/3))+9/(2*pi)*sin(pi*abs(s)/3);

A=zeros(n,n);
b_vect=zeros(n,1);

for i=1:n

    b_vect(i,1)=(integral(g,a+(i-1)*h,a+i*h))/sqrt(h);

    for j=1:n
        A(i,j)=(integral2(K,a+(j-1)*h,a+j*h,a+(i-1)*h,a+i*h))/h;

    end
```

```
end
%faccio SVD di A
s=svd(A);
[U,S,V]=svd(A);

%perturbo b con un vettore le cui componenti sono generate con una
%distribuzione Gaussiana di media 0 e std^2=1e-8
mu=0;
sigma=1e-8;
rng("default")
R=mvnrnd(mu,sigma,n);
b_vect=b_vect+R;

%discretizzo per creare il grafico
x=1:10000;
y=1:10000;

for k=1:10000
    lambda=10^(-6+k*6.5*10^(-4));
    [x(k),y(k)]=norme_lambda(U,s,V,b_vect,n,lambda);

end
%calcolo i valori U^T*b_vect e li metto in z per tracciare il grafico
z=zeros(n,1);
for it=1:n
    z(it)=abs((U(:,it))'*b_vect);


end
%dal grafico sembra che il valore di lambda ottimale è circa 6.8e-4, lo
%confermo con un cerchietto nero che effettivamente si trova nei pressi
%dello spigolo della L
lambda=6.8*10^(-4);
[lambda2,lambda1]=norme_lambda(U,s,V,b_vect,n,lambda);

figure(1)
semilogy(1:n,s,"r.",1:n,z,"bo",1:n,z./s,"kx");

figure(2)
semilogx(y,x,"g.",lambda1,lambda2,"ko");
```

Codice 2

```
function [norma_x,norma_r]= norme_lambda(U,s,V,b_vect,n,lambda)
epsilon=10^(-4);
```

```
x_lambda_err=zeros(n,1);
r_lambda_err=zeros(n,1);
x_lambda=zeros(n,1);
r_lambda=zeros(n,1);
for it=1:n
    x_lambda_err=x_lambda_err+s(it)/(s(it)^2+lambda^2)*V(:,it);
    r_lambda_err=r_lambda_err+lambda^2/(s(it)^2+lambda^2)*U(:,it);
    x_lambda=x_lambda+s(it)/(s(it)^2+lambda^2)*(((U(:,it))')*b_vect)*V(:,it);
    r_lambda=r_lambda+lambda^2/(s(it)^2+lambda^2)*(((U(:,it))')*b_vect)*U(:,it);
end
x_lambda_err=x_lambda_err*epsilon;
r_lambda_err=r_lambda_err*epsilon;
norma_x=norm(x_lambda+x_lambda_err);
norma_r=norm(r_lambda+r_lambda_err);
```

Codice 3

```
tempi=1:5;
esponente=1:5;
epsilon=1e-6;
quantita=1:5;
lambda=6.8e-4;
for numero=1:5
    n=2^(numero+5);
    esponente(numero)=n;
    pi=3.14159265358979323846264338327950288419716939937510582097494459230781640628;
a=-6;
b=6;
h=(b-a)/n;
x_lambda=zeros(n,1);


K= @(x,s) (1+cos(pi*(s-x)/3)).*(abs(s-x)<=3);
g= @(s) (6-abs(s)).*(1+0.5*cos(pi*s/3))+9/(2*pi)*sin(pi*abs(s)/3);

A=zeros(n,n);
b_vect=zeros(n,1);
B=zeros(n,n);
for i=1:n

   b_vect(i,1)=(integral(g,a+(i-1)*h,a+i*h))/sqrt(h);

    for j=1:n
        A(i,j)=(integral2(K,a+(j-1)*h,a+j*h,a+(i-1)*h,a+i*h))/h;
```

```
    end
end
tic
[U,S,V]=svd(A);
s=svd(A);
for it=1:n
x_lambda=x_lambda+s(it)/(s(it)^2+lambda^2)*(((U(:,it))')*b_vect)*V(:,it);
end
tempi(numero)=toc;
quantita(numero)=sum(s/s(1)>epsilon);
s(1) %serve a vedere come si comporta il valore singolare più alto
end
figure(1)
loglog(esponente,tempi,"r-");

figure(2)
plot(esponente,quantita,"k-");

figure(3)
plot(esponente,quantita./esponente,"b-");
```