Esercitazione 3

Esercizio 1 - Teorema di Schoenberg-Whitney

Sia N la dimensione dello spazio per l'intervallo $[x_0, x_n]$. Si può interpolare da $s(t) = \sum_{j=1}^N a_j B_{j,k}(t)$ a siti $\tau_1 < \tau_2 < \dots < \tau_N$ se e solo se $\tau_j \in (\text{nodo}_j, \text{nodo}_{j+k})$ (nodi estesi ordinati da 1 a $N + k - 1$) (questo è il teorema di Schoenberg-Whitney).

Si consideri il seguente esempio per illustrare le condizioni di Schoenberg-Whitney, per $k = 4, n = 5$ e $N = n - k - 1 = 8$ siti scelti in maniera differente per mostrarne il comportamento.

```
% Esercizio per illustrare le condizioni di Schoenberg-Whitney
%
k=4;          % l'ordine (grado = k-1)
n=5;          % nodi t_0,t_1,...,t_n
N=n+k-1;     % la dimensione dello spazio

% t e' la successione di nodi nell'intervallo [t_0,t_n] con (k-1) nodi
% aggiunti all'inizio e altri (k-1) alla fine
t=[-.1*ones(1,(k-1)),linspace(0,1,n+1),1.1*ones(1,k-1)];

% Scelte differenti dei siti x per N=8
%
x=[0,.1,.2,.4,.6,.8,.9 1]; % siti di interpolazione not-a-knot
%x=[.1,.2,.5,.6,.7,.8,.9,1]; % condizioni valide ma interpolazione non buona
%x=[.1,.2,.3,.4,.5,.6,.7,.8] % condizioni non valide

% Costruzione della matrice T per controllare se le condizioni di S-W sono
% realizzate, x(i) dovrebbe essere dentro intervallo (t(i),t(i+k))
% l'ultima colonna di T e' 1 se la condizione e' valida e 0 se non lo e'.
%
T=[];
for i=1:N,
    T=[T; x(i) t(i) t(i+k) (t(i)<x(i) & x(i)<t(i+k))];
end
T
%
% Costruzione della matrice di interpolazione [B_{j,k}(x_i)]
%
M=zeros(N,N);
```

```

for j=1:N
    M(:,j)=BSplineEval(j,k,t,x)';
end
M
%
% interpolazione di una funzione in questi siti
%
f=@(x)sin(2*pi*x);
y=f(x);
c=M\('y');      % coefficienti dell'interpolante
u=linspace(0,1,1000);
s=zeros(size(u));
for j=1:N
    s=s+c(j)*BSplineEval(j,k,t,u);
end
plot(u,s,'r',u,f(u),'b',x,f(x),'o',t,0,'g*')
legend('interpolante','funzione','siti','nodi')

```

```

function value = BSplineEval(i,k,t,u)
% INPUT
% i indice della B-Spline (da 1 a length(t)-k)
% k l'ordine della B-spline
% t vettore dei nodi
% u vettore in cui viene valutata la B-Spline
%
% OUTPUT
% vettore rappresenta i valori della B-Spline in u
%
pp=bspline(t(i:(i+k)));
value=ppval(pp,u);
z=ones(size(u));
j=(u>t(i+k) | u<t(i));
z(j)=0;
value=value.*z;

```

Si confrontino i risultati ottenuti per le diverse scelte dei siti e si forniscano esempi simili per $k = 2, 5$ sostituendo alla condizione not-a-knot, che una condizione per le spline cubiche, un esempio di buona interpolazione.

Esercizio 2

Si consideri il seguente esempio sui minimi quadrati discreti per la ricostruzione, utilizzando B-spline, di una funzione a partire da una serie di dati affetti da rumore.

```
% Esercizio di minimi quadrati usando B-spline
%
close all
clear all
f=@(x)sin(2*pi*x);
x=linspace(0,1,100); % numeri di punti di campionamento
rumore=.1*randn(size(x));
y=f(x)+rumore; % y e' un campione di f con rumore
plot(x,y,'r',x,f(x),'b')
figure
k=4; % l'ordine (grado = k-1)
n=5; % t_0,t_1,...,t_n
N=n+k-1; % dimensione dello spazio

t=[-.1*ones(1,(k-1)),linspace(0,1,n+1),1.1*ones(1,k-1)];
M=zeros(length(x),N);
for j=1:N
    M(:,j)=BSplineEval(j,k,t,x)';
end
c=M\(y'); % soluzione Matlab nel senso di minimi quadrati
u=linspace(0,1,1000); % modello discreto di [0,1]
s=zeros(size(u));
for j=1:N
    s=s+c(j)*BSplineEval(j,k,t,u);
end
plot(u,f(u),'b',u,s,'r',t,0,'g*')
```

Si provi il codice precedente con diverse condizioni di rumore e diversa distribuzione di nodi. Si utilizzi poi una funzione differente, con diverse scelte di rumore e nodi.