

```

f = @(x)(x.^2.*sin(x).*cos(x));
alfa=-pi; beta=pi;

figure(1)
fplot(f,[alfa,beta],'r--');
hold on
title('polinomio interpolante')
xlabel('variabile x')
ylabel('valore della funzione ')
pause

fprintf('Approssimazione mediante interpolazione\n')

for n=[ 6 8 10 12 14]

% Calcolo dei nodi
x=linspace(alfa,beta,n+1)'; % nodi di interpolazione
plot(x,f(x),'*');

% Calcolo dei coeff. del polinomio interpolante
y = f(x);
a = get_polyn(x,y);

% Grafico del polinomio interpolante
t = linspace(alfa,beta,1000); % punti di valutazione per il grafico
yp= polyval(a,t);
plot(t,yp,'k');

% Stima dell'errore
t = linspace(alfa,beta,10000); % punti di valutazione per l'errore
Errore = max( abs(f(t)-polyval(a,t)) );
disp([n,Errore])

    pause

end

hold off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% pb minimi quadrati

figure(2)
fplot(f,[alfa,beta],'r--');
hold on
title('polinomio approssimante con minimi quadrati')
xlabel('variabile x')
ylabel('valore della funzione ')
pause

fprintf('Approssimazione mediante minimi quadrati\n')

n=14;
for m=[ 6 8 10 12]

    % m=n-2;

% Calcolo dei nodi
x=linspace(alfa,beta,n+1)'; % nodi di interpolazione
plot(x,f(x),'*');

% Calcolo dei coeff. del polinomio approssimante
y = f(x);

```

```

    a = get_polyn_ls(x,y,m);

% Grafico del polinomio approssimante
    t = linspace(alfa,beta,1000); % punti di valutazione per il grafico
    yp= polyval(a,t);
    plot(t,yp,'k');

% Stima dell'errore
    t = linspace(alfa,beta,10000); % punti di valutazione per l'errore
    Errore = max( abs(f(t)-polyval(a,t)) );
    disp([n,m,Errore])

    pause

end

hold off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CUT HERE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function a = get_polyn(x,y);
%function a = get_polyn(x,y);
%
% Calcolo dei coeff polinomio interpolante, via matrice Vandermonde

n=length(x)-1;
V(1:n+1,n+1)=ones(n+1,1);
for j=n:-1:1
    V(1:n+1,j) = (x).*V(1:n+1,j+1);
end
fprintf('Numero di condiz V: %d\n',cond(V))
a = V\y;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CUT HERE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function a = get_polyn_ls(x,y,m);
%function a = get_polyn_ls(x,y,m);
%
% Calcolo dei coeff polinomio minimi quadrati, via matrice Vandermonde

n=length(x)-1;
V(1:n+1,m+1)=ones(n+1,1);
for j=m:-1:1
    V(1:n+1,j) = (x).*V(1:n+1,j+1);
end
fprintf('Numero di condiz V: %d\n',cond(V))
a = V\y;
end

```