Esercizio 1

Esercizio

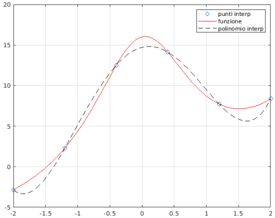
Si consideri la funzione $f(x) = x + e^x + \dfrac{20}{1+x^2} - 5$ *ristretta all'intervallo* $[-2, 2]$. *Facendo anche uso di Matlab*

1. *Determinare il polinomio d'interpolazione di grado* 5 *in forma di Lagrange sui nodi equispaziati* $x_k = -2 + kh, \; k = 0, \ldots, 5$. *Fare un plot comparativo della funzione e del polinomio interpolante.*
2. *Calcolare l'errore d'interpolazione in* $x = 1/2$.
3. *Ripetere i calcoli usando i punti di Chebyshev.*

Soluzione Per rispondere alla prima richiesta, si tratta di usare Matlab per fare il plot comparativo, usando la funzione `lagrai_target.m` fornita nelle slide della Lezione 17 (vedi slide successiva)

Soluzione Esercizio 1

```
clear all; close all;
a=-2; b=2;   % estremi intervallo interpolazione
n=6; % numero dei punti d' interpolazione
f=@(x) x+exp(x)+20./(1+x.^2)-5; % funzione da interpolare
x=linspace(a,b,n); %nodi d' interpolazione equispaziati
z=linspace(a,b,10*n);  %nodi di valutazione

y=f(x);  %funzione nei nodi d'interpolazione
L=[];
for i=1:n,
  L(:,i)=lagrai_target(x,z',i);
end

p=L*y'; % polinomio nei punti di valutazione;

plot(x,f(x),'o', z,f(z), 'r-', z, p','k--');  % plot di confronto
legend('punti interp', 'funzione', 'polinomio interp')
```



Continua Esercizio 1

Per l'errore semplicemente, sempre usando Matlab, si calcola l' errore $r_5(1/2) = f(1/2) - p_5(1/2)$. Usando la relazione $p_5(1/2) = \sum_{i=0}^{5} l_i(1/2) f(x_i)$ sapendo che per valutare $l_i(1/2)$ si può chiamare `lagrai_target(x,1/2,i)` con $i = 0, \ldots, 5$, basta aggiungere il codice

```
p=0;
for i=1:6,
    p=p+lagrai_target(x,0.5,i)*f(x(i));
end
err=f(1/2)-p
```

ottenendo il risultato $err = -0.4534$.

Per rispondere alla terza richiesta, basta sostituire la riga che definisce i punti equispaziati con quella che definisce i punti di Chebyshev mappati in $[-2, 2]$. Ovvero la riga `x=linspace(a,b,n);` con `x=-2*cos(((2*[1:n]-1)*pi)/(2*n));`. L' errore che si ottiene è $err = 9.7e-2$.

Esercizio 2

Esercizio

Calcolare con carta e penna, la costante di Lebesgue sui punti equispaziati $\{0, 0.5, 1\}$.

Soluzione. I polinomi elementari di Lagrange sono $l_0(x) = 2(x-1/2)(x-1)$, $l_1(x) = -4x(x-1)$, $l_2(x) = 2x(x-1/2)$ e la funzione di Lebesgue è $\lambda_2(x) = \sum_{i=0}^{2} |l_i(x)|$. Per ottenere il massimo dividiamo l'analisi come segue

(i) $x \in [0, 1/2]$. Si ha
$\lambda_2(x) = 2(x-1/2)(x-1) - 4x(x-1) - 2x(x-1/2) = -4x^2 + 2x + 1.$

(ii) $x \in [1/2, 1]$. Si ha
$\lambda_2(x) = -2(x-1/2)(x-1) - 4x(x-1) + 2x(x-1/2) = -4x^2 + 6x - 1.$

Dobbiamo ora calcolare il valore della costante di Lebesgue $\Lambda_2 = \max\left\{\max_{x\in[0,1/2]} \lambda_2(x), \max_{x\in[1/2,1]} \lambda_2(x)\right\}$. I punti dove si assumono i due massimi sono $x = 1/4$ e $x = 3/4$. Inoltre $\lambda_2(1/4) = \lambda_2(3/4) = 5/4 = 1.25$. Pertanto $\Lambda_2 = 1.25$.

Continua Esercizio 2: grafici

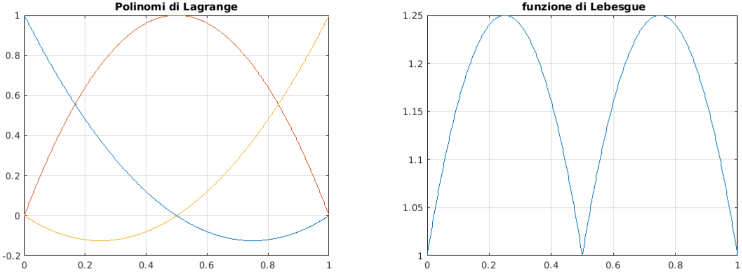


Figure: sx: polinomi elementari di Lagrange; dx: funzione di Lebesgue

Esercizio 3

Esercizio

Calcolare le differenze divise di ordine 3 della funzione $f(x) = \sqrt{2+x^2}$ *nei punti* $\{0, 1, 1, 2\}$ *(arrotondare il risultato a 2 cifre decimali). Nota bene: la differenza divisa di ordine 1 di* $f[x_i, x_i] = f'(x_i)$ *se f derivabile.*

La tabella delle differenze divise è la seguente

x_i	y_i	ordine 1	ordine 2	ordine 3
0	$\sqrt{2}$			
1	$\sqrt{3}$	$\sqrt{3}-\sqrt{2}$		
1	$\sqrt{3}$	$\frac{\sqrt{3}}{3}$	$-\frac{2}{3}\sqrt{3}+\sqrt{2}$	
2	$\sqrt{6}$	$\sqrt{6}-\sqrt{3}$	$-\frac{4}{3}\sqrt{3}+\sqrt{6}$	$\frac{\sqrt{6}-\sqrt{2}-\frac{2}{3}\sqrt{3}}{2} \approx -0.06$

Table: Differenze divise della funzione $\sqrt{x^2+2}$

Esercizio 4

Esercizio

Si consideri la funzione $f(x) = \dfrac{20}{1+\log(x^2)} - 5\,\sin(e^x)$ *ristretta all'intervallo* $[1,2]$*. Determinare l'unico polinomio (d'interpolazione) di secondo grado,* $p_2(x) = a_0 + a_1 x + a_2 x^2$ *che soddisfa alle condizioni d'interpolazione generalizzate*

$$p_2(1) = f(1),\;\; p_2(2) = f(2),\;\; \int_1^2 p_2(x)dx = \int_1^2 f(x)dx\ .$$

Per il calcolo dell'integrale definito della funzione usare la funzione Matlab `quadl`*, chiamandola come segue Q = quadl(FUN,A,B) (con tolleranza di default* $1.e-6$*). Fare quindi il grafico della funzione, del polinomio e di* $\|f - p_2\|_\infty$.

Soluzione Esercizio 4

Soluzione. Detto $p_2(x) = a_0 + a_1 x + a_2 x^2$ il polinomio di grado 2 che cerchiamo, per determinarne i coefficienti un modo è risolvere il sistema $Ma = c$ con matrice

$$M = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3/2 & 7/3 \end{bmatrix}$$

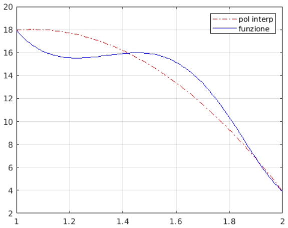
e termine noto

$$c = \begin{bmatrix} f(1) \\ f(2) \\ \int_1^2 f(x)dx \end{bmatrix} = \begin{bmatrix} 20 - 5\sin(e) \\ \frac{20}{1+\log(4)} - 5\sin(e^2) \\ q \end{bmatrix}.$$

con `q=quadl(f,1,2)`. Calcolando i valori (arrontondando a 2 decimali) si ha `c=[17.95; 3.91; 13.56]`. Risolvendo il sistema si ottengono i coefficienti $a = M\backslash c = [0.43; 33.3; -15.78]$.

Continua Esercizio 4: script e grafico di confronto

```
clear all; close all;

f=@(x) 20./(1+log(x.^2))-5*sin(exp(x)); % funzione

M=[1 1 1; 1 2 4; 1 3/2 7/3]; % Matrice sistema
c=[17.95; 3.91; 13.56]; % termine noto
a=M\c;  %coefficienti del polinomio

x=linspace(1,2, 100); % punti di valutazione

p=polyval(flip(a),x); % valuto il polinomio in x
% devo fare il flip dei coefficienti per la solita ragione
errore=norm(p-f(x),inf)  % errore in norma infinito

plot(x,p,'r', x,f(x), 'b-');  %plot comparativo
legend('pol interp', 'funzione');
```



con *errore* ≈ 2.17

Esercizio 5

Esercizio

Si consideri la funzione $f(x) = \cos(x^3)\,(x - 2\,\pi)\,e^{-x},\ x \in [0, \pi]$. *Sperimentalmente si determini il grado del polinomio d'interpolazione, costruito sui nodi di Chebsyhev in* $[0, \pi]$, *che approssima* $f(x)$ *in norma infinito a meno di* $tol = 1.e - 4$.

Soluzione. Si tratta di determinare n cosicché $\max_{x\in[0,\pi]} |f(x) - p_n(x)| \leq 10^{-4}$ con p_n costruito usando i punti di Chebyshev. Lo possiamo fare usando lo script visto nell' Esercizio 1 modificandolo opportunamente (vedi slide successiva)

Continua soluzione Esercizio 5: script e risultati

```
clear all;  close all;
a=0; b=pi;   % estremi intervallo interpolazione

% funzione da interpolare
f=@(x) cos(x.^3).*(x-2*pi).*exp(-x);
z=linspace(a,b,500);  %nodi di valutazione
fz=f(z); % funzione nei punti di valutazione
err=1; n=1; % nr punti d'interpolazione iniziali
while err >1.e-4
L=[];
x=(a+b)/2-(b-a)/2*cos(((2*[1:n]-1)*pi)/(2*n));  % nodi di Chebyshev ordinati in [0,pi]
y=f(x);  %funzione nei nodi d' interpolazione
for i=1:n,
  L(:,i)=lagrai_target(x,z',i);
end
  p=L*y'; % polinomio nei punti di valutazione;
  % calcolo dell' errore
  for i=1:length(p)
    err1(i)=abs(fz(i)-p(i));
  end
  err=max(err1);
  n=n+1;
end
disp('grado del polinomio richiesto')
n-1
disp(' errore = ')
err
plot(x,f(x),'o', z,fz, 'r-', z, p','k--');  % plot di confronto
legend('punti interp', 'funzione', 'polinomio interp')
```

Plot finale

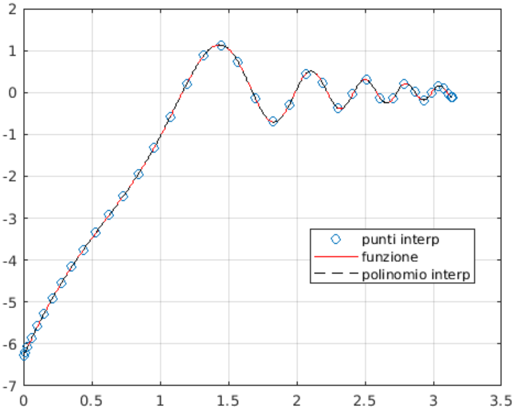


Figure: Per ottenere un errore minore di 1e-4, serve un polinomio di grado 39. L'errore calcolato è $err = 4.5e - 5$ (arrotondato)

Algoritmo DD

Per determinare i coefficienti b_i, $i = 0, \ldots, n$ della forma di Newton dell' interpolante si usa l'algoritmo delle differenze divise il cui codice Matlab è descritto nella funzione

```
function d=DD(x,y)
%---------------------------------------------
%inputs: x=vettore dei punti; y=vettore dei valori f(x) (DD di ordine 0)
%output: d=vettore delle DD
%----------------------------------------------
n=length(x);
d=y;
for i=2:n
  for j=2:i
     d(i)=(d(i)-d(j-1))/(x(i)-x(j-1));
  end
end
```

Il vettore `d` viene usato per memorizzare le DD da quella di ordine 0 a quella di ordine n.

Osservazione

Vantaggio della forma di Newton

Per passare da p_n a p_{n+1} serve calcolare solo un coefficiente in più, d_{n+1} la differenza divisa di ordine $n+1$. Infatti

$$\boxed{p_{n+1}(x) = p_n(x) + d_{n+1} \prod_{i=0}^{n} (x - x_i)}$$

Vediamo un esempio nella prossima slide

Esempio

Consideriamo i punti $\{0, 2, 3, 5\}$ e $f(x) = e^{x-1}$. La tabella delle differenze divise è la seguente

x_i	y_i	ordine 1	ordine 2	ordine 3
0	$e^{-1} = d_0$			
2	e	$\frac{e-e^{-1}}{2} = d_1$		
3	e^2	$e^2 - e$	$\frac{e^2-2e+e^{-1}}{3} = d_2$	
5	e^4	$\frac{e^4-e^2}{2}$	$\frac{e^4-3e^2+2e}{6}$	$\frac{e^4-6e^2+8e-3e^{-1}}{30} = d_3$

Table: Differenze divise della funzione e^{x-1}

Pertanto $p_2(x) = d_0 + d_1(x-0) + d_2(x-0)(x-2)$ mentre
$p_3(x) = p_2(x) + d_3(x-0)(x-2)(x-3)$

Schema di Hörner

Serve per valutare il polinomio p_n in forma di Newton in un punto x con meno operazioni additive e moltiplicative

Schema di Hörner

```
p=d_n;
for i=n-1:-1:1
   p= (x-x_i)*p+d_i;
end
```

Esempio

Consideriamo il polinomio $p_2(x) = a_0 + a_1 x + a_2 x^2$. La valutazione in un punto x richiede 2 addizioni e 3 moltiplicazioni.

Con Hörner, il polinomio viene scritto come segue
$p_2(x) = (a_2 x + a_1)x + a_0$ riducendo a 2 sia le addizioni che le moltiplicazioni.

Polinomio d'interpolazione in forma di Newton: codice

```
n=...;  %grado del polinomio d'interpolazione
N=....; %nr. punti di valutazione
a=...; b=...; % estremi intervallo interpolazione
f=@(x)....; % funzione da interpolare
x=....; y=f(x);
d=DD(x,y);  % calcolo le DD
z=linspace(a,b,N);  %punti di valutazione
% applico lo schema di Hoerner per valutare il polinomio
% su tutti i punti z
for k=1:N
  p(k)=d(n);
  for i=n-1:-1:1
     p(k)= (z(k)-x(i))*p(k)+d(i);
  end
end
```

Esercizio. Usare questo codice per interpolare $f(x) = \cos(x^2)$, $x \in [-\pi, \pi]$ al variare di n.