Prova di esonero relativa al capitolo 4

Esercizio 1. Dimostrare l'esistenza ed unicità del polinomio interpolante una funzione su $n+1$ ascisse distinte.

Esercizio 2. Calcolare il polinomio interpolante la funzione $\cos(x)$ nelle ascisse $x_i = i \cdot \pi/2$, $i = 0, 1, 2$, nella sua forma di Lagrange.

Esercizio 3. Calcolare il polinomio interpolante di Hermite la funzione $\cos(x)$ nelle ascisse 0 e $\pi/2$.

Esercizio 4. Scrivere una function Matlab che, dati in ingresso un intero `n` e due reali `a` e `b`, calcoli le ascisse di Chebyshev per valutare il polinomio interpolante di grado `n` sull'intervallo `[a,b]`.

Esercizio 5. Scrivere una function Matlab che, dati in ingresso i vettori `xi` e `fi` con i dati di interpolazione, ed il vettore `x` in cui valutarlo, calcoli efficientemente il polinomio interpolante mediante la sua forma di Newton.

Esercizio 6. Definire cosa è una spline cubica su una partizione assegnata; specificare quante condizioni sono necessarie per individuarne una specifica; definire, infine, quali sono le condizioni che individuano la spline cubica <u>interpolante</u> naturale sui nodi della partizione assegnata.

Punteggio esercizi: esercizi 1–5, punti 8 cadauno; esercizio 6, punti 20.

Esercizio 1. $\exists!\, p \in \Pi_n : p(x_i) = f_i,\ i=0,\dots,n,$ con $x_0 < x_1 < \dots < x_n$.

Se pongo $p(x) = \sum_{k=0}^{n} a_k x^k$, le condizioni di interpolazione si possono scrivere come

$$\sum_{k=0}^{n} a_k x_i^k = f_i,\quad i=0,\dots,n,$$

ovvero come

$$\begin{bmatrix} x_0^0 & x_0^1 & \cdots & x_0^n \\ \vdots & \vdots & & \vdots \\ x_n^0 & x_n^1 & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f_0 \\ \vdots \\ f_n \end{bmatrix},$$

che è un sistema lineare, la cui matrice dei coefficienti è la trasposta di una matrice di Vandermonde, definita da ascisse distinte e, pertanto, nonsingolare. La soluzione del sistema lineare quindi esiste ed è unica.

Esercizio 2. $x_0 = 0,\ x_1 = \frac{\pi}{2},\ x_2 = \pi,\ f_0 = 1,\ f_1 = 0,\ f_2 = -1$. Pertanto il polinomio interpolante, nella sua forma di Lagrange, sarà dato da $p(x) = L_{02}(x) - L_{22}(x)$, con

$$L_{02}(x) = \frac{(x-\frac{\pi}{2})(x-\pi)}{\frac{\pi^2}{2}},\qquad L_{22}(x) = \frac{x(x-\frac{\pi}{2})}{\frac{\pi^2}{2}}.$$

Esercizio 3. $x_0 = 0,\ x_1 = \pi/2,\ f_0 = 1,\ f_0' = 0,\ f_1 = 0,\ f_1' = -1$.

	0	1	2	3
0	1			
0	1	0		
$\frac{\pi}{2}$	0	$-\frac{2}{\pi}$	$-\frac{4}{\pi^2}$	
$\frac{\pi}{2}$	0	-1	$\frac{2}{\pi}\left(\frac{2}{\pi}-1\right)$	$\frac{4}{\pi^2}\left(\frac{4}{\pi}-1\right)$

$\Rightarrow$ $f[0] = 1$, $f[0,0] = 0$,

$f[0,0,\frac{\pi}{2}] = -\frac{4}{\pi^2}$

$f[0,0,\frac{\pi}{2},\frac{\pi}{2}] = \frac{4}{\pi^2}\left(\frac{4}{\pi}-1\right)$

$$\Rightarrow p(x) = 1 - \frac{4}{\pi^2}x^2 + \frac{4}{\pi^2}\left(\frac{4}{\pi}-1\right)x^2\left(x-\frac{\pi}{2}\right).$$

Esercizio 4.

```
function x = ceby(a,b,n)
% x = ceby(a,b,n)   calcola le ascisse di Chebyshev
%                   per il polinomio di interpolazione
%                   di grado n su [a,b].
%
if a>=b || n~=fix(n) || n<0, error('dati errati'); end
x = (a+b)/2 + ((b-a)/2)*cos((2*[0:n]+1)*(pi/(2*n+2)));
return
```

Esercizio 6. Sia $\Delta = \{a = x_0 < x_1 < \dots < x_n = b\}$ una partizione assegnata di un intervallo $[a,b]$. Diremo che $s_3(x)$ è una spline cubica su Δ se:

a) $s_3|_{[x_{i-1},x_i]}(x) \in \Pi_3,\ i=1,\dots,n$; b) $s_3 \in C^{(2)}[a,b]$;

Per individuare univocamente una spline cubica sulla partizione Δ occorrono $n+3$ condizioni (indipendenti).

Le condizioni che individuano la spline cubica naturale interpolante una data funzione $f(x)$ sulla partizione Δ sono:

i) $\underbrace{s_3(x_i) = f(x_i),\ i=0,\dots,n}_{n+1 \text{ condizioni}}$; ii) $\underbrace{s_3''(a) = 0 = s_3''(b)}_{2 \text{ condizioni}}$.

Esercizio 5.

```
function   y = newton (xi, fi, x)
%    y = newton (xi, fi, x)     calcola il polinomio interpolante
%                               di Newton sulle coppie (xi, fi), nelle
%                               ascisse contenute nel vettore x.
%
%

n = length(xi);                                % controllo correttezza dati
if n<=0 || n~=length(fi)
    error('dati inconsistenti');
end
for i=1:n-1
    for j=i+1:n
        if (xi(i)==xi(j))
            error('ascisse non distinte');
        end
    end
end
                  % calcolo differenze divise
c = fi;
for i = 1:n-1
    for j = n:-1:i+1
        c(j) = (c(j) - c(j-1)) / (xi(j) - xi(j-i));
    end
end                               % calcolo del polinomio
y = c(n) * ones(size(x));
for i = n-1:-1:1
    y = y.*(x- xi(i)) + c(i);
end
return.
```