ESAME CALCOLO NUMERICO PROVA DI LABORATORIO
LAUREA IN INFORMATICA
TERZO APPELLO 23/08/2022

Consegna Compito: saranno visibili solo i files consegnati in tempo tramite moodle. Consegnare anche i files del docente. Il caricamento dei files deve avvenire con supervisione del docente: **in caso constrario si è esclusi dalla prova**.

Tempo di svolgimento: 90 minuti.

Supponiamo che $f : [x_{min}, x_{max}] \to [y_{min}, y_{max}]$ sia una funzione continua strettamente crescente. Esiste allora una funzione $g : [y_{min}, y_{max}] \to [x_{min}, x_{max}]$ tale per cui $f(g(y)) = y$ per ogni $y \in [y_{min}, y_{max}]$ e $g(f(x)) = x$ per ogni $x \in [x_{min}, x_{max}]$ detta *inversa di f*. Cercheremo di costruire per interpolazione un polinomio p_n che approssimi g nota f basandoci solo su valutazioni di f.

Esercizio 1 (25 p.ti)**.** Si scriva una function avente la chiamata

`xeval=InverseFunctionPoly(f,xmin,xmax,toll,n,yeval)`

che, dati `f` function handle della funzione che vogliamo invertire, `xmin,xmax` estremi di un intervallo ove `f` è continua e crescente, `toll` tolleranza per il metodo di bisezione (si veda più sotto),`n` grado polinomiale da utilizzare, e `yeval` vettore di punti di valutazione, calcoli le valutazioni $p_n(y_1^{eval}), \ldots, p_n(y_m^{eval})$ del polinomio $p_n \approx g$ che interpola i dati $(y_i^{interp}, x_i^{interp})$ con $i = 1, 2, \ldots, n+1$ (cioè $p_n(y_i^{interp}) = x_i^{interp}$), dove gli y_i^{interp} sono punti di Chebyshev-Lobatto nell'intervallo

`[ymin,ymax]=[f(xmin),f(xmax)]`,

e gli x_i^{interp} sono soluzione approssimate di $f(x) = y_i^{interp}$ calcolate tramite bisezione. L'algoritmo di calcolo deve essere il segunte:

(1) Calcolare `ymin` e `ymax` valutando f e costruire un vettore colonna `yinterp` di $n + 1$ punti di Chebyshev-Lobatto nell'intervallo `[ymin,ymax]`. Si ricorda che, nell'intervallo $[-1, 1]$ i punti di Chebyshev Lobatto sono `cos((0:n)'/n*pi)`.
(2) Calcolare (una per una all'interno di un ciclo for) le componenti del vettore `xinterp`, dove `xinterp(i)` è soluzione approssimata di `f(x)-yinterp(i)=0`: usare la function `Bisezione.m` fornita dal docente utilizzando gli estremi dell'intervallo `[xmin,xmax]`, `method='s'` e tolleranza `toll`. **Consiglio importante:** conviene ad ogni iterazione del ciclo `for` definire un'opportuna anonymous function `fi` di cui calcolare lo zero.
(3) Utilizzando la funzione `LagrangePoly.m` fornita dal docente si creino i polinomi di Lagrange dei nodi di interpolazione `yinterp` valutati sui punti di valutazione `yeval` e il vettore contenente le valutazioni di p_n, ossia $(p_n(y_1^{eval}), \ldots, p_n(y_{end}^{eval}))^t$ (con opportuno prodotto matrice-vettore).

Esercizio 2 (6 p.ti)**.** Si crei uno script `Esercizio2.m` che definisca $f(x) = e^x$, ponga l'intervallo di interesse $[0, 5]$ e crei 100 punti equispaziati y^{eval} di valutazione nell'immagine di questo intervallo tramite f. Calcoli la valutazione sui punti y^{eval} dell'approssimazione p_n dell'inversa g di f tramite

`InverseFunctionPoly`, si usi a tal fine `n=8` e `toll=1e-10`. Poi venga creata un' unica figura con i grafici di p_n e g (l'inversa vera) corredata da legeda.