ESERCIZI DA SVOLGERE PER L' ELABORATO
CAP 4
CN A.A. 16/17

1. Scrivere una function Matlab che, presi in input due vettori $\mathbf{x}$ e $\mathbf{f}$ di $n+1$ componenti e preso un vettore di tabulazione $xval$ di $nval$ componenti, fornisce in output un vettore $Pval$ di $nval$ componenti tale che $Pval_j$ contiene il valore assunto in $xval_j$ dal polinomio di grado $\leq n$ interpolante le coppie $(x_i, f_i), i = 1, \ldots, n+1$ (usare la forma di Newton, richiamare la function per il calcolo delle differenze divise e usare l'algoritmo di Horner per la valutazione).

2. Testare il funzionamento della precedente function per interpolare la funzione $f(x) = \frac{1}{1+x^2}$ nell'intervallo $[-5\,,\,5]$. Utilizzare, per n crescente (pari) con $n \leq 20$, sia ascisse equispaziate che ascisse generalizzate di Chebyshev e confrontare i risultati (fare una tabella con l'errore per i vari valori di n e confrontare il grafico di $f(x)$ con quello del polinomio interpolante per qualche valore di n). Ripetere il test con la funzione $f(x) = x \sin x$ nell'intervallo $[0\,,\pi]$ (fare uno script per organizzare la sperimentazione).

3. Scrivere una function Matlab che, presi in input due vettori $\mathbf{x}$ e $\mathbf{f}$ di $n+1$ componenti, con $\mathbf{x}$ ad elementi distinti ed f_i che indica il valore assunto in x_i da una funzione $f(x)$, fornisce in output il vettore dei momenti $\mathbf{M}$ di $n+1$ componenti relativo alla spline cubica naturale interpolante nei nodi (usare l'algoritmo di fattorizzazione LU specifico per una tridiagonale a diagonale dominante e espandere $\mathbf{M}$ inserendovi una prima e ultima componente nulle) e i vettori $\mathbf{r}$ e $\mathbf{q}$ di n componenti necessari per costruire in forma polinomiale a tratti la spline cubica s_3 naturale avente $\mathbf{x}$ come nodi e tale che $s_3(x_i) = f_i, i = 1, \ldots, n+1$.

4. Scrivere una function Matlab che, presi in input tre vettori $\mathbf{x}, \mathbf{f}$ e $\mathbf{M}$ di $n+1$ componenti e preso un vettore di tabulazione $xval$ di $nval$ componenti, fornisce in output un vettore $sval$ di $nval$ componenti tale che $sval_j$ contiene il valore assunto in $xval_j$ da una spline cubica s_3 avente $\mathbf{x}$ come vettore dei nodi e tale che $s_3(x_i) = f_i, i = 1, \ldots, n+1$ e $s_3''(x_i) = M_i, i = 1, \ldots, n+1$.

5. Testare le precedenti functions per interpolare con una spline naturale la funzione $f(x) = \frac{1}{1+x^2}$ nell'intervallo $[-5\,,\,5]$. Utilizzare, per n crescente (pari) con $n \leq 20$, ascisse equispaziate e confrontare i risultati con quelli ottenuti con la spline cubica not–a–knot definita mediante il comando `spline` (fare una tabella con l'errore per i vari valori di n e confrontare il grafico della spline naturale con quello di $f(x)$ e con quello della spline not–a–knot per qualche valore di n). Ripetere il test con la funzione $f(x) = x \sin x$ nell'intervallo $[0\,,\pi]$ (fare uno script per organizzare la sperimentazione).

6. FAC/GR3

   Scrivere una function gemella di quella costruita per il punto 3, che fornisce in output i momenti della spline cubica periodica (usare lo specifico algoritmo di fattorizzazione LU).

7. FAC/GR3

   Testare la precedente function per interpolare con una curva spline parametrica chiusa una sequenza di punti assegnati nel piano $\mathbf{P}_i\,,\ i = 1, \ldots, n$, con $\mathbf{P}_n = \mathbf{P}_1$. Utilizzare, a tale scopo dei parametri $0 = t_1 < \ldots < t_n = 1$ tali che $(t_{i+1} - t_i)$ sia proporzionale alla distanza fra $\mathbf{P}_i$ e $\mathbf{P}_{i+1}$.

8. Costruire una function Matlab che calcola il polinomio di migliore approssimazione di grado $m < n$ delle coppie $(x_i, y_i), i = 1, \ldots, n+1$, dopo aver verificato che esistono almeno $m+1$ ascisse distinte (in caso contrario fornisce un messaggio di errore) e il corrispondente valore della funzione obiettivo minimizzata. A tale scopo assemblare la matrice di Vandermonde rettangolare e utilizzare il `backslash`.

9. Sperimentare la function precedente mediante uno script in cui si definiscono i dati per due tests: per entrambi si pone $n = 10$ e si definiscono le ascisse equispaziate in $[-1\,,\,1]$ ma nel primo si pone $m = 1$, e si definiscono le corrispondenti ordinate ponendo $y_i = 5x_i + 2 + \epsilon\gamma_i$; nel secondo invece si pone $m = 2$ e $y_i = 3x_i^2 + 2x_i + 1 + \epsilon\gamma_i$, dove γ_i è un numero random in $[0\,,\,1]$ e ϵ una costante positiva (usare 0.1 e 0.2).

10. Per il primo test dell'esercizio precedente confrontare la retta ottenuta con quella che si ottiene considerando la y come variabile indipendente.