LABORATORIO 4

Argomenti: Approssimazione, interpolazione polinomiale

Algoritmo difdiv

Input: x, y, n
Output: a

for $i = 1, \ldots, n$
    for $j = n+1, \ldots, i+1$
        $y_j \leftarrow (y_j - y_{j-1})/(x_j - x_{j-i})$
    end
end
$a \leftarrow y$

Algoritmo interp

Input: x, a, z, n
Output: p

$p \leftarrow a_{n+1}$
for $i = n, \ldots, 1$
    $p \leftarrow p(z - x_i) + a_i$
end

1. Generare un vettore x di $n = 2, 3, \ldots, 20$ elementi equispaziati nell'intervallo $[0, 1]$ (estremi inclusi) e calcolare, tramite il comando Matlab `vander`, la matrice di Vandermonde V_n associata al vettore x. Successivamente, calcolare e rappresentare graficamente il condizionamento in norma 1, 2 e infinito della matrice di Vandermonde.

   Infine, definito il vettore b in modo tale che il sistema lineare $V_n x = b$ ammetta come soluzione il vettore unitario, risolvere il sistema lineare per $n = 2, \ldots, 20$ e rappresentare graficamente l'errore relativo associato alla soluzione rispetto all'ordine del sistema. Commentare i risultati.

2. Generare un vettore x di $n = 2, 3, \ldots, 20$ elementi equispaziati nell'intervallo $[0, 1]$ (estremi inclusi) e definire la funzione $f(x) = \sin(\pi x)$. Mediante i comandi Matlab `polyfit` e `polyval`, calcolare e rappresentare graficamente il polinomio interpolante i dati $(x_i, f(x_i))$, $i = 1, \ldots, n$. Alla luce dei risultati ottenuti nell'esercizio precedente, commentare i risultati.

3. Calcolare e rappresentare graficamente i polinomi di grado $n = 5, 10, 15, 20$ interpolanti la funzione di Runge $f(x) = \frac{1}{1+x^2}$ in nodi equispaziati nell'intervallo $[-5, 5]$. Commentare i risultati.

   Successivamente ripetere l'esercizio per l'intervallo $[1, 2]$. Commentare i risultati.

   Infine ripetere l'esercizio per l'intervallo $[-5, 5]$ e per i nodi di Chebichev

$$z_i = \frac{b-a}{2} \cos\left(\frac{(2i-1)\pi}{2(n+1)}\right) + \frac{b+a}{2}, \; i = 1, \ldots, n+1.$$

   Commentare i risultati.

4. Implementare in due *m-file* di tipo *function*, denominati `difdiv.m` e `interp.m`, gli algoritmi per il calcolo dei coefficienti e la valutazione del polinomio di Newton interpolante i dati (x_i, y_i), $i = 1, \ldots, n+1$, rispettivamente.

   Strutturare la *function* `difdiv` in modo tale che, ricevendo in input i vettori x ed y contenenti le ascisse e le ordinate dei punti di interpolazione, restituisca in output il vettore a contenente i coefficienti del polinomio interpolante di Newton, ovvero le differenze divise $f[x_1]$, $f[x_1, x_2]$, $\ldots, f[x_1, x_2, \ldots, x_n]$. Strutturare la *function* `interp` in modo tale che, ricevendo in input i vettori x, a e z, quest'ultimo contenente i punti in cui si vuole valutare il polinomio, restituisca in output il vettore contenente le valutazioni del polinomio interpolante.

   Successivamente costruire la tabella delle differenze divise associata ai seguenti dati

x_i	$f(x_i)$
1.8	1.341641
1.9	1.378405
2.0	1.414214
2.1	1.449138
2.2	1.483240

   e il corrispondente polinomio di Newton, per approssimare la funzione $f(x) = \sqrt{x}$ nell'intervallo $[1.8, 2.2]$. Tracciare il grafico del polinomio interpolante e della funzione $f(x)$ ed esaminare il comportamento dell'errore nei punti $x = 1.85, \; 1.95, \; 2.05, \; 2.15$.

5. L'algoritmo per la costruzione della tabella delle differenze divise, implementato nella *function* `difdiv` dell'esercizio precedente, consta di due cicli, uno interno all'altro. Utilizzando opportunamente operazioni di tipo vettoriale in luogo di quelle di tipo scalare, è possibile eliminare il ciclo più interno?

6. Utilizzare la *function* `polyfit` di MATLAB per determinare i coefficienti della somma esponenziale di ordine 3

$$f(x) = c_1 e^{3x} + c_2 e^{2x} + c_3 e^x + c_4$$

   passante per i punti $(0, 1)$, $(1, 4)$, $(2, 8)$ e $(3, 16)$. Successivamente rappresentare graficamente la funzione f e i punti di interpolazione.