Calcolo Scientifico, A.A. 2022/23
Appello 19 luglio 2023

Tutte le function sono scaricabili dalla pagina `paola-gervasio.unibs.it/CS/MATLAB`.

Esercizio 1.
Si consideri la matrice

$$A = \begin{bmatrix} 10^5 & 5 & 2 & 1 \\ 5 & 10 & -4 & 0 \\ 2 & -4 & 8 & -1 \\ 1 & 0 & -1 & 2 \end{bmatrix}$$

Punto 1.1 Analizzare le proprietà della matrice.

Punto 1.2 Scrivere una function MATLAB che costruisca l'inversa della matrice A richiamando una fattorizzazione opportuna che sfrutti le proprietà della matrice. Quindi verificare la correttezza della function confrontando il proprio risultato con quello fornito dal comando `inv` di MATLAB.

Punto 1.3 Poiché gli unici errori che si introducono sono quelli dell'aritmetica di macchina, stimare il massimo errore commesso nel calcolo degli elementi di A^{-1}.

Esercizio 2.
Per disegnare l'arco di circonferenza descritto dalla funzione

$$f(x) = \sqrt{1 - x^2}$$

con $x \in [-0.8, 0.8]$, viene costruito l'interpolatore composito lineare $p_1^c(x)$ che interpola $f(x)$ in N (≥ 2) punti nell'intervallo $[-0.8, 0.8]$.

Punto 2.1 Determinare per via teorica* il minimo numero N di punti equispaziati che permetta di disegnare la curva con un errore

$$\|f - p_1^c\|_\infty \leq 10^{-3},$$

sapendo che la costante C che entra in gioco nella stima dell'errore è $C = 0.125$.
Con l'N così calcolato, costruire e disegnare $p_1^c(x)$ sull'intervallo $[-0.8, 0.8]$.

Punto 2.2 Verificare numericamente che l'errore $\|f - p_1^c\|_\infty$ tende a zero all'aumentare del numero di nodi (equispaziati), con un ordine di convergenza come predetto dalla teoria. A tale scopo, calcolare l'interpolatore p_1^c al variare di N da 10 a 1000 con passo 10, valutare l'errore e plottarlo rispetto all'ampiezza H degli intervallini.

Punto 2.3 Cosa possiamo concludere se, invece di prendere $x \in [-0.8, 0.8]$, si considerasse $x \in [-1, 1]$?

---

*Per il calcolo delle derivate di f si può utilizzare il toolbox di calcolo simbolico di Matlab. Esempio per calcolare la derivata prima di una funzione g: `syms x; g=sin(x); g1=diff(g,x) % (funzione simbolica); fun1=matlabFunction(g1) % (function handle)`

Domanda 1.

Il metodo di Newton:

1. a cosa serve,
2. come si formula,
3. che proprietà di convergenza ha,
4. come si generalizza a sistemi di equazioni non lineari,
5. qual è il costo computazionale di ogni iterazione del metodo di Newton per risolvere sistemi di equazioni non lineari.

Domanda 2.

I metodi di Eulero (in avanti e all'indietro) per approssimare equazioni differenziali ordinarie:

1. loro formulazione per equazioni scalari,
2. come si ottengono i due metodi,
3. proprietà di convergenza e stabilità,
4. confronto computazionale fra i due metodi.