PROGETTO 1: INTERPOLAZIONE POLINOMIALE CON NODI DI LEJA APPROSSIMATI

1. Contesto e fine

Sia $I = [a, b] \subset \mathbb{R}$, definiamo la seguente sequenza di punti

- Sia ξ_0 un punto di I. Supponiamo ora che siano già stati definiti s punti $\xi_0, \ldots, \xi_{s-1}$, allora il punto successivo ξ_s è scelto in modo da massimizzare la funzione
$$\xi \to |\det(VDM(\xi_0, \ldots, \xi_{k-1}, \xi)|,$$
dove $VDM(\xi_0, \ldots, \xi_s)$ è la matrice di Vandermonde costruita con i punti $\xi_0, \ldots, \xi_s$.

Tale sequenza di punti è denominata sequenza di Leja e l'interpolante costruita su tali punti gode di eccellenti proprietà di stabilità e convergenza (se la funzione interpolata è sufficientemente regolare).

Massimizzare sull'intervallo continuo è però computazionalmente costoso, quindi è preferibile utilizzare una loro approssimazione andando a cercare il punto su una discretizzazione dell'intervallo ovvero su una mesh discreta abbastanza fitta su I.

Questo permette di scegliere il s-esimo punto non come quello che massimizza il determinante della Vandermonde in tutto l'intervallo ma, se X_M è la discretizzazione di I, il punto è calcolato come
$$x_s = \arg \max_{\xi \in X_M} |\det(VDM(\xi_0, \ldots, \xi_{s-1}, \xi)|.$$

Si noti che il determinante della matrice di Vandermonde ha la seguente proprietà ricorsiva. Se $\xi_0, \ldots, \xi_{s-1}, \xi_s$ sono i nodi di interpolazione, allora
$$\det(VDM(\xi_0, \ldots, \xi_s)) = \det(VDM(\xi_0, \ldots, \xi_{s-1}) \prod_{i=0}^{s-1} (\xi_s - \xi_i).$$

Questa riscrittura permette di ricercare il punto x_s senza effetturare il calcolo del determinante di tutta la matrice di tipo Vandermonde. Ma x_s corrisponde al punto che massimizza la produttoria, ovvero
$$x_s = \arg \max_{x \in X_M} \prod_{i=0}^{s-1} |x - \xi_i|.$$

Similmente, tale estrazione di punti può essere implementata attraverso una fattorizzazione LU con pivoting della matrice di tipo Vandermonde.

Infatti se $V \in \mathbb{R}^{M \times s+1}$ è la matrice di tipo Vandermonde di grado s utilizzante i polinomi di Chebyshev, ovvero con $V_{i,j} = \cos((j-1)\arccos(x_i))$, $i = 1, \ldots, M$, $j = 1, \ldots, s+1$ relativa ai punti $X_M = \{x_1, \ldots, x_M\}$, allora i primi $s + 1$ elementi di X_M ordinati secondo l'ordinamento dato dalla permutazione corrispondente alla matrice P della fattorizzazione $PA = LU$, equivale a cercare il massimo iterativamente lasciando invariante il valore assoluto dei sottodeterminanti.

Tale procedimento risulta meno costoso computazionalmente all'aumentare del grado.

Vogliamo quindi sviluppare un programma in grado di calcolare la costante di Lebesgue del polinomio che utilizza come nodi una approssimazione dei punti di Leja.

- Calcolando dei punti di Leja approssimati partendo da una mesh di 10^5 o 10^4 punti equidistanti in $I = [-1, 1]$ usando la strategia descritta precedentemente;

- Calcolando la costante di Lebesgue come il massimo della funzione di Lebesgue

$$\lambda_n(x) = \sum_{i=0}^{n} |\ell_i(x)|$$

con ℓ_i i polinomi di Lagrange, su una discretizzazione dell'intervallo I.

2. Function richieste

2.1. Calcolo dei punti di Leja approssimati.

Algoritmo 1 per i punti discreti di Leja

Vogliamo creare una function `DLP` avente chiamata `dlp=DLP(x,d)` con parametri di input

- `x` vettore di punti nell'intervallo I da cui estrarre i punti di Leja approssimati;
- `d` scalare intero positivo che corrisponde al grado relativo al polinomio interpolante di cui estraiamo i $d+1$ nodi;

ed output

- `dlp` vettore di $d+1$ punti corrispondenti ai punti di Leja approssimati.

Tale function deve estrarre, partendo dal primo elemento del vettore x, i successivi punti di Leja approssimati massimizzando la produttoria vista precedentemente.

Algoritmo 2 per i punti discreti di Leja

Vogliamo creare una function `DLP2` avente chiamata `dlp=DLP2(x,d)` con parametri di input

- `x` vettore di punti nell'intervallo I da cui estrarre i punti di Leja approssimati;
- `d` scalare intero positivo che corrisponde al grado relativo al polinomio interpolante di cui estraiamo i $d+1$ nodi;

ed output

- `dlp` vettore di $d+1$ punti corrispondenti ai punti di Leja approssimati.

Tale function deve estrarre, partendo dal primo elemento del vettore x, i successivi punti di Leja approssimati. Questi però dovranno utilizzare la fattorizzazione LU di matlab. E come matrice di Vandermonde si chiede di usare quella costruita con la base di Chebyshev.

2.2. Calcolo della costante di Lebesgue.

Vogliamo creare una function `leb_con` avente chiamata `L = leb_con(z,x)` con parametri di input

- `z` vettore riga dei nodi dell'interpolante su cui costruire la funzione di Lebesgue;
- `x` vettore colonna di punti nell'intervallo I su cui valutare la funzione di Lebesgue;

ed output

- `L` scalare reale, che è l'approssimazione della costante di Lebesgue.

La function dovrà utilizzare massimo un ciclo e dovrà valutare il valore assoluto dei polinomi di Lagrange sul vettore `x`, sommandoli e andando poi a selezionare il massimo tra questi valori.

3. Sperimentazione

Per sperimentare i programmi sviluppati si creino al variare del grado del polinomio interpolante $d = 1, \dots, 50$, i punti di Leja approssimati con entrambi gli algoritmi controllando i tempi computazionali utilizzando le funzioni `tic` e `toc` di Matlab di entrambi gli algoritmi. Si faccia un grafico al crescere del grado dei tempi computazionali.

Per mostrare la stabilità del polinomio interpolante si crei il grafico della costante di Lebesgue in scala semilogaritmica dei punti di Leja approssimati.

Infine per ciascun n si testi l'interpolante costruita su tali nodi (si usi solo l'algoritmo più efficiente) con la funzione

$$f(x) = \frac{1}{x - 1.3},$$

paragonandola con l'interpolante costruita usando come nodi i punti equispaziati. Si costruisca l'interpolante senza l'utilizzo della funzione `polyfit` di Matlab ma costruendo la matrice di Vandermonde che utilizza come base polinomiale i polinomi di Chebyshev e ricavando i coefficienti risolvendo il sistema lineare dato dalle condizioni di interpolazione.