1. Fondamenti dell'Interpolazione Polinomiale con Punti di Leja

1.1 Definizione e Proprietà dei Punti di Leja

I punti di Leja costituiscono una sequenza speciale di nodi per l'interpolazione polinomiale. Data una sequenza $\xi_0, \xi_1, \ldots, \xi_{s-1}$, il punto successivo ξ_s viene scelto massimizzando:

$$\xi_s = \operatorname{argmax}\{\xi \in I\} \, |\det(VDM(\xi_0,\ldots,\xi_{s-1},\xi))|$$

dove VDM rappresenta la matrice di Vandermonde.

1.2 Significato Geometrico

La massimizzazione del determinante della matrice di Vandermonde ha un significato geometrico profondo:

- Garantisce che i punti siano distribuiti in modo da minimizzare il mal condizionamento del problema
- Evita il clustering dei punti, un problema comune nell'interpolazione
- Produce una sequenza annidata (nested) di punti ottimali

2. Implementazione Pratica

2.1 Approccio con Discretizzazione

Per ragioni computazionali, invece di cercare il massimo su tutto l'intervallo continuo [a,b], si utilizza una discretizzazione XM:

$$x_s = \operatorname{argmax}\{x \in XM\} \, |\det(VDM(\xi_0,\ldots,\xi_{s-1},x))|$$

2.2 Proprietà Ricorsiva Fondamentale

Il determinante della matrice di Vandermonde ha una proprietà ricorsiva cruciale:

$$\det(VDM(\xi_0,\ldots,\xi_s)) = \det(VDM(\xi_0,\ldots,\xi_{s-1})) \prod_{i=0}^{s-1} (\xi_s - \xi_i)$$

Questa proprietà permette di:

- Semplificare il calcolo del determinante
- Ridurre la complessità computazionale

- Implementare l'algoritmo DLP in modo efficiente

3. Metodi di Implementazione

3.1 Metodo DLP (Direct Leja Points)

Basato sulla massimizzazione diretta della produttoria:

$$x_s = \text{argmax}_{\{x \in XM\}} \prod_{i=0}^{s-1} |x - \xi_i|$$

Vantaggi:

- Implementazione diretta e intuitiva
- Buone prestazioni per gradi bassi
- Facilità di debug e manutenzione

3.2 Metodo DLP2 (LU-based Leja Points)

Utilizza la fattorizzazione LU della matrice di Vandermonde in base Chebyshev:

$$V[i,j] = \cos((j-1)\arccos(x_i))$$

Vantaggi:

- Più efficiente per gradi elevati
- Migliore stabilità numerica
- Sfrutta le ottimizzazioni delle librerie numeriche

4. Costante di Lebesgue e Stabilità

4.1 Definizione della Costante di Lebesgue

La costante di Lebesgue Λ_n è definita come:

$$\Lambda_n = \max_{\{x \in [a,b]\}} \sum_{i=0}^{n} |\ell_i(x)|$$

dove $\ell_i(x)$ sono i polinomi fondamentali di Lagrange.

4.2 Importanza per la Stabilità

La costante di Lebesgue:

- Misura la stabilità dell'interpolazione
- Fornisce un limite superiore all'errore relativo

- Indica quanto l'interpolante è sensibile agli errori nei dati

4.3 Comportamento per Diversi Nodi

- Punti equidistanti: crescita esponenziale ($\sim 2^n/n!$)
- Punti di Chebyshev: crescita logaritmica (~log(n))
- Punti di Leja: comportamento intermedio vantaggioso

5. Base di Chebyshev

5.1 Definizione

I polinomi di Chebyshev di prima specie sono definiti come:

```
Tn(x) = cos(n arccos(x))
```

5.2 Vantaggi nell'Uso della Base di Chebyshev

- Migliore condizionamento numerico
- Riduzione delle oscillazioni spurie
- Distribuzione ottimale degli zeri
- Naturale connessione con la trasformata discreta del coseno

6. Analisi Sperimentale

6.1 Confronto dei Tempi Computazionali

Si deve analizzare:

- Scalabilità con il grado del polinomio
- Efficienza relativa di DLP vs DLP2
- Overhead computazionale per gradi diversi

6.2 Test di Accuratezza

Usando la funzione test f(x) = 1/(x-1.3):

- Confronto con interpolazione su nodi equidistanti
- Analisi dell'errore di interpolazione
- Verifica della stabilità numerica

6.3 Metriche di Valutazione

- Tempo di esecuzione

- Costante di Lebesgue
- Errore massimo di interpolazione
- Condizionamento numerico del sistema

7. Considerazioni Implementative

7.1 Gestione della Precisione Numerica

- Uso di aritmetica a doppia precisione
- Gestione attenta del calcolo dei determinanti
- Normalizzazione appropriata dei valori

7.2 Ottimizzazioni Possibili

- Memorizzazione dei risultati intermedi
- Parallelizzazione del calcolo per mesh grandi
- Utilizzo di strutture dati efficienti

7.3 Criteri di Stop

- Tolleranza sul massimo della produttoria
- Numero massimo di iterazioni
- Controllo della stabilità numerica

8. Conclusioni e Best Practices

8.1 Quando Usare Quale Metodo

- DLP: per gradi bassi ($n < 20$) o implementazioni didattiche
- DLP2: per gradi elevati o applicazioni che richiedono alta efficienza

8.2 Limitazioni da Considerare

- Costo computazionale per mesh molto fini
- Precisione numerica per gradi molto elevati
- Comportamento vicino a singolarità della funzione

8.3 Estensioni Possibili

- Implementazione in domini multidimensionali
- Adattamento a funzioni con singolarità
- Integrazione con altri metodi numerici