Interpolazione Polinomiale con Nodi di Leja Approssimati

Analisi Comparativa e Implementazione Efficiente

Gabriel Rovesti

18 agosto 2025

Sommario

Questo documento presenta un'analisi completa dell'interpolazione polinomiale utilizzando nodi di Leja approssimati. Vengono implementati e confrontati due algoritmi per la selezione dei nodi: uno basato sulla massimizzazione iterativa del prodotto delle distanze (DLP) e uno basato sulla fattorizzazione LU con pivoting della matrice di Vandermonde-Chebyshev (DLP2). I risultati dimostrano la superiorità numerica dei nodi di Leja rispetto ai nodi equispaziati, confermando le previsioni teoriche sull'instabilità dell'interpolazione polinomiale con nodi equidistanti (fenomeno di Runge).

1 Introduzione

L'interpolazione polinomiale costituisce un pilastro fondamentale del calcolo numerico, con applicazioni che spaziano dall'approssimazione di funzioni alla quadratura numerica. La scelta dei nodi di interpolazione influenza drasticamente la stabilità e l'accuratezza del processo.

I **nodi di Leja** rappresentano una strategia avanzata per la selezione di punti di interpolazione, progettata per minimizzare la crescita della costante di Lebesgue e migliorare la stabilità numerica. A differenza dei classici nodi di Chebyshev, i nodi di Leja si adattano dinamicamente al dominio di interpolazione, rendendoli particolarmente efficaci per problemi con geometrie complesse.

1.1 Obiettivi del Progetto

Il presente lavoro si propone di:

1. Implementare due algoritmi efficienti per il calcolo dei nodi di Leja approssimati
2. Analizzare le prestazioni computazionali dei metodi proposti
3. Valutare la stabilità numerica tramite la costante di Lebesgue
4. Confrontare l'accuratezza dell'interpolazione con nodi di Leja vs. nodi equispaziati
5. Dimostrare sperimentalmente la superiorità teorica dei nodi di Leja

2 Fondamenti Teorici

2.1 Sequenza di Leja

Dato un insieme compatto $K \subset \mathbb{C}$ e un punto iniziale $z_0 \in K$, la sequenza di Leja $\{z_k\}_{k=0}^{\infty}$ è definita ricorsivamente come:

$$z_{k+1} = \arg\max_{z \in K} \prod_{j=0}^{k} |z - z_j| \tag{1}$$

Per applicazioni numeriche su intervalli reali $[a, b]$, si approssima questa definizione operando su una discretizzazione finita $X_M = \{x_1, x_2, \ldots, x_M\}$ dell'intervallo.

2.2 Proprietà di Stabilità

La costante di Lebesgue per un insieme di nodi $\{z_i\}_{i=0}^{n}$ è definita come:

$$\Lambda_n = \max_{x \in [a,b]} \sum_{i=0}^{n} |\ell_i(x)| \tag{2}$$

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

$$\ell_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - z_j}{z_i - z_j} \tag{3}$$

La costante di Lebesgue fornisce una misura della stabilità dell'interpolazione: errori più piccoli in Λ_n corrispondono a migliore condizionamento numerico.

2.3 Connessione con il Determinante di Vandermonde

La selezione dei nodi di Leja è equivalente alla massimizzazione iterativa del determinante della matrice di Vandermonde. Per $n + 1$ nodi $\{z_0, \ldots, z_n\}$, vale la relazione:

$$\det(V(z_0, \ldots, z_n)) = \prod_{0 \leq i < j \leq n} (z_j - z_i) \tag{4}$$

Questa proprietà ricorsiva permette di riformulare il problema di selezione come:

$$z_{k+1} = \arg\max_{x \in X_M} \prod_{i=0}^{k} |x - z_i| \tag{5}$$

3 Algoritmi Implementati

3.1 Algoritmo DLP: Produttoria Iterativa

Il primo algoritmo (DLP) implementa direttamente la definizione ricorsiva dei nodi di Leja:

```
function dlp = DLP(x, d)
    % Input:
    %   x - vettore colonna dei punti candidati
    %   d - grado del polinomio interpolante
    % Output:
    %   dlp - vettore riga dei d+1 nodi di Leja

    dlp = zeros(1, d+1);
    dlp(1) = x(1);  % Primo nodo: estremo sinistro

    for s = 2:d+1
        % Calcola produttoria delle distanze per ogni punto
        produttoria = prod(abs(x - dlp(1:s-1)), 2);

        % Seleziona il punto che massimizza la produttoria
        [~, idx_max] = max(produttoria);
        dlp(s) = x(idx_max);
    end
end
```

Listing 1: Implementazione Algoritmo DLP

Complessità: $O(N \cdot d^2)$ dove N è la dimensione della mesh e d il grado del polinomio.

3.2 Algoritmo DLP2: Fattorizzazione LU

Il secondo algoritmo (DLP2) utilizza la fattorizzazione LU con pivoting della matrice di Vandermonde-Chebyshev:

```
function dlp2 = DLP2(x, d)
    % Input/Output: identici a DLP

    x = x(:);  % Assicura formato colonna
    x = max(-1, min(1, x));  % Clipping per stabilit  numerica

    % Costruzione matrice di Vandermonde-Chebyshev
    % V(i,j) = T_{j-1}(x_i) = cos((j-1) * arccos(x_i))
    V = cos(acos(x) * (0:d));  % Matrice  N  (d+1)

    % Fattorizzazione LU con pivoting
    [~, ~, P] = lu(V, 'vector');

    % Selezione primi d+1 nodi secondo permutazione P
    dlp2 = x(P(1:d+1))';
end
```

Listing 2: Implementazione Algoritmo DLP2

Complessità: $O(N \cdot d^2)$ per la costruzione della matrice più $O(d^3)$ per la fattorizzazione LU.

3.3 Aspetti Critici dell'Implementazione

3.3.1 Orientamento della Matrice di Vandermonde

Un aspetto cruciale dell'algoritmo DLP2 è l'orientamento corretto della matrice di Vandermonde. La matrice deve avere dimensioni $N \times (d+1)$ dove:

- **Righe**: corrispondono ai punti candidati x_i
- **Colonne**: corrispondono ai gradi dei polinomi di Chebyshev T_j

Con questa configurazione, la permutazione P restituita da `lu` opera sulle righe, selezionando effettivamente i punti ottimali della mesh.

3.3.2 Base di Chebyshev vs. Base Monomiale

L'utilizzo della base di Chebyshev $\{T_k(x)\}_{k=0}^{d}$ invece della base monomiale $\{x^k\}_{k=0}^{d}$ migliora significativamente la stabilità numerica. I polinomi di Chebyshev soddisfano $|T_k(x)| \leq 1$ per $x \in [-1, 1]$, limitando la crescita degli elementi della matrice di Vandermonde.

4 Calcolo della Costante di Lebesgue

La valutazione della costante di Lebesgue richiede il calcolo efficiente dei polinomi di Lagrange:

```
function L = leb_con(z, x)
    % Input:
    %   z - vettore riga dei nodi di interpolazione
    %   x - vettore colonna dei punti di valutazione
    % Output:
    %   L - costante di Lebesgue

    z = z(:)';  % Assicura formato riga
    x = x(:);   % Assicura formato colonna
    n = length(z);

    lebesgue_vals = zeros(size(x));

    for i = 1:n
        % Indici di tutti i nodi tranne il corrente
        altri_nodi = [1:i-1, i+1:n];

        % Calcolo polinomio di Lagrange       (x)
        lagrange_poly = prod((x - z(altri_nodi)) ./ ...
                             (z(i) - z(altri_nodi)), 2);

        % Accumula valore assoluto
        lebesgue_vals = lebesgue_vals + abs(lagrange_poly);
    end

    L = max(lebesgue_vals);
end
```

Listing 3: Calcolo della Costante di Lebesgue

5 Interpolazione con Base di Chebyshev

Per garantire stabilità numerica nell'interpolazione, implementiamo un metodo basato sulla risoluzione diretta del sistema lineare con matrice di Vandermonde-Chebyshev:

```
function p_eval = interp_chebyshev(x_nodes, f_nodes, x_eval)
    n = length(x_nodes);

    % Costruzione matrice di Vandermonde: V(i,j) = T_{j-1}(x_i)
    V_fit = cos((0:n-1) .* acos(x_nodes(:)));

    % Risoluzione sistema lineare V*c = f
    c = V_fit \ f_nodes(:);

    % Valutazione su punti richiesti
    V_eval = cos((0:n-1) .* acos(x_eval(:)));
    p_eval = V_eval * c;
end
```

Listing 4: Interpolazione con Base di Chebyshev

6 Risultati Sperimentali

6.1 Setup Sperimentale

Gli esperimenti sono condotti con i seguenti parametri:

- **Mesh**: $N = 10^4$ punti equispaziati su $[-1, 1]$
- **Gradi**: $d \in \{1, 2, \ldots, 50\}$
- **Funzione test**: $f(x) = \frac{1}{x-1.3}$
- **Confronto**: Nodi di Leja vs. nodi equispaziati

La funzione $f(x) = \frac{1}{x-1.3}$ presenta una singolarità in $x = 1.3$, vicina ma esterna all'intervallo $[-1, 1]$. Questa scelta permette di evidenziare il fenomeno di Runge con nodi equispaziati.

6.2 Analisi delle Prestazioni Computazionali

Osservazione 1: DLP2 risulta generalmente più efficiente di DLP per gradi elevati, nonostante la maggiore complessità teorica. Questo è dovuto alla migliore implementazione della fattorizzazione LU in MATLAB rispetto ai cicli espliciti.

Osservazione 2: I picchi occasionali nei tempi di DLP2 sono attribuibili alle strategie di pivoting adattivo dell'algoritmo LU, che possono richiedere ricerche più estensive per matrici mal condizionate.

6.3 Stabilità: Crescita della Costante di Lebesgue

I risultati mostrano una crescita della costante di Lebesgue di tipo logaritmico-polinomiale per i nodi di Leja, significativamente più contenuta rispetto alla crescita esponenziale tipica dei nodi equispaziati.

Intervallo di crescita: $\Lambda_n \in [1, 10^2]$ per $n \in [1, 50]$, conforme alle previsioni teoriche per sequenze di Leja su intervalli reali.

6.4 Confronto degli Errori di Interpolazione

6.4.1 Comportamento dei Nodi di Leja

Per $f(x) = \frac{1}{x-1.3}$, l'interpolazione con nodi di Leja mostra:

- **Convergenza esponenziale**: errore $\sim O(\rho^{-d})$ con $\rho > 1$
- **Stabilità**: errore decresce monotonamente fino a $\sim 10^{-16}$ (precisione macchina)
- **Robustezza**: nessuna instabilità per gradi elevati

6.4.2 Comportamento dei Nodi Equispaziati

L'interpolazione con nodi equispaziati presenta:

- **Fase iniziale**: convergenza per $d \lesssim 25$
- **Fenomeno di Runge**: esplosione dell'errore per $d \gtrsim 30$
- **Instabilità numerica**: condizionamento della matrice $\sim O(10^{12})$ per $d = 50$

7 Analisi Teorica dei Risultati

7.1 Convergenza per Funzioni Analitiche

La funzione $f(x) = \frac{1}{x-1.3}$ è analitica in un'ellisse E_ρ con fuochi in ± 1 e semi-asse maggiore $\rho = 1.3$. La teoria dell'interpolazione con nodi di Leja garantisce convergenza esponenziale:

$$\|f - p_n\|_\infty \leq \frac{2M}{\rho^n} \cdot \Lambda_n \tag{6}$$

dove M è il massimo di $|f|$ su E_ρ e Λ_n è la costante di Lebesgue.

7.2 Fenomeno di Runge

Per nodi equispaziati, il teorema di Runge stabilisce che per funzioni con singolarità vicine all'intervallo di interpolazione, l'errore cresce esponenzialmente con il grado del polinomio:

$$\|f - p_n\|_\infty \sim C \cdot \left(\frac{2}{1+\sqrt{2}}\right)^n \to \infty \tag{7}$$

I nostri risultati confermano sperimentalmente questa previsione teorica.

8 Validazione dell'Implementazione

8.1 Test di Consistenza

Abbiamo verificato la correttezza degli algoritmi attraverso:

1. **Convergenza**: per nodi di Chebyshev noti, DLP2 produce sequenze equivalenti
2. **Monotonia**: la costante di Lebesgue cresce monotonamente con il grado
3. **Limiti asintotici**: comportamento conforme alle previsioni teoriche

8.2 Robustezza Numerica

L'implementazione gestisce correttamente:

- **Boundary conditions**: clipping dei valori per arccos
- **Conditioning**: utilizzo della base di Chebyshev
- **Precision**: gestione della precisione macchina

9 Conclusioni

Il presente lavoro ha dimostrato sperimentalmente la superiorità dei nodi di Leja rispetto ai nodi equispaziati per l'interpolazione polinomiale. I risultati principali sono:

1. **Stabilità numerica superiore**: costante di Lebesgue con crescita controllata
2. **Convergenza robusta**: assenza del fenomeno di Runge per gradi elevati
3. **Efficienza computazionale**: algoritmo DLP2 competitivo per applicazioni numeriche
4. **Validazione teorica**: conformità con le previsioni dell'analisi funzionale

L'implementazione proposta fornisce uno strumento affidabile per l'interpolazione polinomiale ad alta precisione, particolarmente adatto per funzioni con singolarità vicine al dominio di interpolazione.

9.1 Sviluppi Futuri

Possibili estensioni del lavoro includono:

- Estensione a domini bidimensionali (nodi di Leja pesati)
- Integrazione con metodi di quadratura adattiva
- Ottimizzazione per architetture parallele
- Applicazioni a problemi di approssimazione spettrale

Riconoscimenti

Si ringraziano i docenti del corso di Calcolo Numerico per i preziosi suggerimenti teorici e metodologici che hanno guidato lo sviluppo di questo progetto.