
Calcolo Numerico - II

Indice

1 Ancora sugli Autovalori **4**
1.1 Iterazione QR . . . 4
1.2 Ottimizzazione dell'Iterazione QR . . . 7
1.3 Analisi di Stabilità . . . 9
1.4 Risultati di Perturbazione . . . 9

2 Definizioni Generali **12**
2.1 Buona Posizione di un problema dato . . . 12
2.2 Consistenza, Stabilità e Convergenza di un Metodo . . . 13

3 Approssimazione per Interpolazione **16**
3.1 Prime Definizioni . . . 16
3.2 Forma di Lagrange . . . 18
3.3 Analisi dell'Errore . . . 20
3.4 Nodi non Equidistanti . . . 23
3.5 Nodi di Chebyshev . . . 23
3.6 Analisi di Stabilità . . . 28
3.7 Forma di Newton . . . 32

4 Interpolazione Composta **35**
4.1 Stima dell'Errore di Convergenza . . . 38
4.2 Splines . . . 38
4.3 Splines Lineare . . . 39
4.4 Splines Cubiche . . . 40
4.5 Completamento delle Condizioni . . . 42
4.6 Risultati di Convergenza e Regolarità . . . 43
4.7 Risultati di Convergenza . . . 45

5 Approssimazione di Integrali **46**
5.1 Presentazione del Problema . . . 46
5.2 Formula del Rettangolo o del Punto Medio . . . 46
5.3 Formula del Trapezio . . . 48
5.4 Formula di Cavalieri - Simpson . . . 49
5.5 Formule Quadratiche Composte . . . 50
5.6 Stima computazionale dell'ordine di Convergenza . . . 51
5.7 Esercizi degli ultimi capitoli . . . 53
5.8 Formule di Newton - Cotes . . . 60

6 Approssimazione di Derivate (differenze finite) **62**
6.1 Somme in Avanti e in Indietro . . . 62
6.2 Approssimazione di Derivate Seconde . . . 64

7 Equazioni non Lineari **65**
7.1 Presentazione del Problema . . . 65
7.2 Condizionamento del Problema . . . 66
7.3 Metodo di Bisezione . . . 67
7.4 Criterio di Arresto . . . 69

7.5 Metodo di Newton 70
7.6 Analisi di Convergenza di Newton 73
7.7 Esercizi su Newton 75
7.8 Varianti (Metodi Quasi Newton) 77

8 Zeri di Polinomi **79**
8.1 Problema e Prime Soluzioni 79
8.2 Questioni di Stabilità 82

9 Iterazione di Punto Fisso **84**
9.1 Presentazione del problema 84

1 Ancora sugli Autovalori

1.1 Iterazione QR

In questa prima parte l'idea è di finire quello che era stato appena iniziato alla fine del modulo scorso. Andiamo a capire cosa c'è dietro alla funzione `eig(A)`.
Prima di proseguire facciamo un richiamo.

Definizione 1.1.1: Decomposizione di Schur

Sia $A \in \mathbb{C}^{n\times n}$, allora esistono $Q \in \mathbb{C}^{n\times n}$ unitaria e $R \in \mathbb{R}^{n\times n}$ triangolare superiore tali che:

$$A = QRQ^H$$

Questo è vero per ogni matrice simmetrica e gli elementi sulla diagonale di R, cioè $R_{i,i}$, contengono gli autovalori di R (generalmente senza ordine).

Noi vogliamo costruire una procedura per arrivare a questa scomposizione. Notiamo che possiamo fare a meno dell'ipotesi di diagonalizzabilità, in quanto è difficile trovare gli autovettori.

L'idea per la **Iterazione QR** (che dal nome si può capire coinvolga la fattorizzazione QR) è quella di determinare una successione $\{T_k\}_{k\in\mathbb{N}}$ con $T_0 = A$ e $T_k = U_k^H A U_k$ con U unitaria tale che:

$$T_k \xrightarrow{k\to+\infty} T$$

In modo che T sia triangolare superiore con tutti gli autovalori di A sulla diagonale principale.

Osservazione. Si richiede che U unitarie in modo che tutte le trasformazioni siano più stabili

"Numericamente" parlando, quello che vogliamo fare è annullare tutti i termini posti sotto alla diagonale principale.
Perché non usiamo Gauss? Perché non abbiamo trasformazioni unitarie, quindi sono trasformazioni non del tutto accurate. *Perché non usiamo la fattorizzazione QR?* Perché con la fattorizzazione QR facciamo trasformazioni solo da destra, e non da entrambe le parti, quindi gli autovalori non è detto che siano gli stessi. Con la fattorizzazione di Schur invece ci assicuriamo di trovare una matrice triangolare simile a quella di partenza.

Andiamo a scrivere una versione base dell'algoritmo di Iterazione QR.

Algoritmo di Iterazione QR

Sia $T_0 = A$
Per $k = 0, 1, ...$
  $T_k = Q_k R_k$
  $T_{k+1} := R_k Q_k$

Osservazione. Per evitare confusioni, facciamo delle osservazioni. All'interno del ciclo i due comandi, seppur incredibilmente simili, sono profondamente diversi. Nel comando sopra noi stiamo facendo la fattorizzazione QR: è infatti nota la matrice T_k e stiamo ricavando $Q_k R_k$. Nel comando sotto invece noi stiamo definendo T_{k+1} (lo si può vedere infatti dall'utilizzo di ":="); qui sono note R_k e Q_k

Vediamo ora come sono collegati queste linee dell'algoritmo:
Dalla prima abbiamo che:

$$T_k = R_k Q_k \qquad \Rightarrow \qquad Q_k^H T_k = R_k$$

Tutto questo funziona, cioè posso calcolare inverse di matrici, in quanto sto utilizzando matrici quadrate in $\mathbb{C}$. Andando poi a sostituire nella seconda linea dell'algoritmo si ha che:

$$T_{k+1} = R_k Q_k = Q_k^H T_k Q_K$$

Cioè abbiamo ottenuto una trasformazione unitaria che rende simili T_k e T_{k+1}, quindi gli autovalori sono necessariamente gli stessi.
Se andiamo a mettere insieme tutte le iterazioni abbiamo che:

$$T_{k+1} = \underbrace{Q_k^H Q_{k-1}^H \cdots Q_0^H}_{U_k^H} T_0 \underbrace{Q_0 \cdots Q_{k-1} Q_k}_{U_k} = U_k^H T_0 U_k$$

Inoltre U_{k+1} è ancora unitaria in quanto prodotto di matrici unitarie.
Ovviamente per $k \to +\infty$ si ottiene che: $T = U^H T_0 U$

Osservazione. Il costo computazionale di ogni iterazione dell'algoritmo è $\mathcal{O}(n^3)$, cioè è estremamente costoso (ad ogni iterazione facciamo una fattorizzazione QR)

Andiamo a vedere dei risultati di convergenza per l'Iterazione QR.

Teorema 1.1.2

Sia $A \in \mathbb{C}^{n \times n}$ con $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$ autovalori semplici in modulo e λ_i autovalori di A (quindi A diagonalizzabile). Allora:

$$\lim_{t \to +\infty} T_k = T$$

Con T matrice triangolare superiore avente sulla diagonale gli autovalori λ_i

Osservazione. Se A è normale, allora T è diagonale.

Considerazioni Aggiuntive. *Abbiamo delle ipotesi restrittive, per le quali λ_i siano semplici in modulo, ma in un certo senso richiama il metodo di convergenza degli autovalori. Infatti questo è un risultato che si basa proprio sulle stesse ipotesi.*
Volendo la convergenza si può esser dimostrata anche con ipotesi meno restrittive.

Per la dimostrazione sarà utile richiamare il metodo delle potenze, anche in una sua forma molto base.

Algoritmo del Metodo delle Potenze

Sia $x_0 \in \mathbb{C}^n$, con norma 1
Per $k = 0, 1, ...$
$y = Ax^{(k)}$
$x^{(x+1)} = \frac{y}{\|y\|}$
$\lambda^{(K+1)} = (x^{(k+1)})^H A x^{(k+1)}$

Quest'idea può essere generalizzata anche al caso di una matrice con ℓ colonne. In tal caso, l'algoritmo prende il nome di **Iterazione del Sottospazio**. In questo caso l'algoritmo diventa

Algoritmo dell'Iterazione del Sottospazio

Sia $U_0 \in \mathbb{C}^{n\times\ell}$, con colonne ortonormali
Per $k = 0, 1, ...$
$Y = AU^{(k)}$
$[U^{(k+1)}, R] = \texttt{QR}(Y)$
$\Lambda^{(K+1)} = (U^{(k+1)})^H AU^{(k+1)}$

Facciamo un minimo di chiarezza su alcune cose che sono state usate.
Quando facciamo $[U^{(k+1)}, R] = \texttt{QR}(Y)$, abbiamo utilizzato la notazione di Matlab, in modo da non creare confusione con $Y = QR$ oppure con $U^{(k+1)}R = Y$. Sostanzialmente quello che è facciamo è fare la fattorizzazione QR (ridotta) della matrice Y. *Utilizzare la fattorizzazione QR non è la sola, volendo di potevano usare anche altre.*
La matrice Λ è una matrice quadrata che sta in $\mathbb{C}^{\ell\times\ell}$, la cui diagonale tende ad approssimare un gruppo di autovalori di A. In particolare approssima gli ℓ autovalori più grandi di A (sempre in modulo).

Per far vedere che l'iterazione QR coincide con l'iterazione del sottospazio basta prendere $\ell = n$. Infatti:

$$T_k = U_k^H AU_k \quad \text{e} \quad T_k = Q_k R_k$$

Mettendo tutto insieme si ha che:

$$Q_k R_k = U_k^H AU_k \in \mathbb{C}^{n\times n}$$

Portando a sinistra U_k si ha che:

$$\underbrace{U_k Q_k}_{U_{k+1}} R_k = AU_k \quad \Rightarrow \quad U_{k+1}R_k = AU_k$$

Che è esattamente quanto fatto nella fattorizzazione QR.

Osservazione. Se gli autovalori non sono distinti, allora troviamo un autospazio

Torniamo un secondo indietro alla fattorizzazione QR. Se per un certo r si ha che $|\lambda_r| = |\lambda_{r+1}|$ e abbiamo gli autovalori sulla diagonale ordinati per modulo, allora si ha che:

$$(T_k)_{(r:r+1,r:r+1)} = \begin{pmatrix} t_{r,r} & t_{r,r+1} \\ t_{r+1,r} & t_{r+1,r+1} \end{pmatrix}$$

Cioè all'interno della matrice c'è un blocco 2×2:

$$T_k = \begin{pmatrix} \ddots & & \\ & \square & \\ & & \ddots \end{pmatrix}$$

In questo caso non possiamo sperare che $t_{r+1,r}$ tenda a zero. Quindi rimarrà un blocco all'interno della matrice. Però i suoi autovalori saranno una approssimazione di λ_r e λ_{r+1}.
Tutto questo è vero anche quando ci sono più blocchi di quella forma e con dimensioni $\ell \times \ell$, non necessariamente 2×2.

L'iterazione QR serve molto per l'approssimazione di basi dell'autospazio. Infatti, supponiamo di avere una matrice con μ auotovalore con molteplicità algebrica 4 per esempio. Il metodo delle potenze mi da solo un autovettore. Se invece prendo una matrice con 4 colonne, ottengo una base di V_μ.

Osservazione. L'iterazione QR di base è molto costosa, con un costo pari a $\mathcal{O}(n^3)$ per iterazione. Inoltre non sapendo neanche quante iterazioni faccia, non sappiamo neanche dare una stima dall'alto. Nel migliore dei modi. Il suo costo computazionale arriverebbe a $\mathcal{O}(n^4)$, un costo incredibilmente eccessivo. Dobbiamo quindi trovare un modo per alleggerire il costo computazionale.

1.2 Ottimizzazione dell'Iterazione QR

Per poter alleggerire il costo computazionale possiamo lavorare sotto due aspetti:

1. Abbassare il costo di ogni iterazioni
2. Abbassare il numero di iterazioni

Cominciamo dal primo punto.
Se T_k fosse di tipo Hessenberg Superiore (Triangolare superiore con la prima diagonale sotto quella principale non nulla), allora la fattorizzazione costerebbe molto di meno $\mathcal{O}(n^2)$. Supponiamo quindi che T_k sia Hessenberg Superiore. Quello che vogliamo fare allora è azzerare tutti gli elementi sotto la diagonale principale. Ci basta utilizzare Givens. *Supponiamo $T_k \in \mathbb{C}^{4\times 4}$ per rappresentare il tutto graficamente*

$$T_k = \begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \end{pmatrix}$$

Posso allora scrivere Q_k matrice di rotazione di Givens e scrivere $Q_k^T T_k = R_k$, che, come detto prima, ha un costo pari a $\mathcal{O}(n^2)$. Quindi si ha che:

$$T_k = Q_k R_k$$

Per creare T_{k+1} ho:

$$T_{k+1} = R_k Q_k = Q_k^H T_k Q_k$$

Questa matrice è ancora di tipo Hesenberg superiore. Quindi quando all'iterazione k-esima ottengo una matrice Hessenberg Superiore, anche tutte quelle successive lo sono.
Facciamo allora in modo che T_0 sia di tipo Hessenberg Superiore. Cioè determiniamo $\hat{Q}_0$ tale che:

$$T_0 = \hat{Q}_0^H A \hat{Q}_0 \text{ di tipo Hessenberg Superiore con } \hat{Q}_0 \text{ unitaria}$$

La nuova iterazione diventa allora:

Secondo Algoritmo di Iterazione QR
Sia $T_0 = \hat{Q}_0^H A \hat{Q}_0$
Per $k = 0, 1, ...$
  $T_k = Q_k R_k$
  $T_{k+1} := R_k Q_k$

Osservazione. Abbiamo trovato gli autovalori, come troviamo gli autovettori?

Per poterli trovare mi basta prendere un vettore $v \in \mathbb{C}^n$ casuale e applicare il metodo delle potenze inverse traslate, in quanto ho già una stima molto buona dell'autovalore, cioè:

$$(A - T_{j,j} I)x = v \qquad T_{j,j} = \sigma$$

Ma questa matrice è singolare, come posso fare? E questo il punto. Numericamente parlando, la matrice $A - T_{j,j}I$ non è singolare, perché $T_{j,j}$ non è esattamente l'autovalore, è solo un'ottima approssimazione. Quella matrice è estremamente vicina all'essere singolare, ma non lo è. Proprio per questo motivo, ci basta una sola iterazione del metodo delle potenze inverse traslate per trovare la direzione giusta.
Con il caso diagonale si ha che:

$$\begin{pmatrix} \lambda_1 - T_{j,j} & & \\ & \ddots & \\ & & \lambda_n - T_{j,j} \end{pmatrix} x = v \qquad \Leftrightarrow x = \begin{pmatrix} \frac{v_1}{\lambda_1 - T_{j,j}} \\ \vdots \\ \frac{v_n}{\lambda_n - T_{j,j}} \end{pmatrix}$$

Se $T_{j,j}$ è molto vicino a λ_k con $k \in \{1, ..., n\}$, allora si ha che:

$$\lambda_k - T_{j,j} \approx 0 \qquad \Rightarrow \qquad \frac{v_k}{\lambda_k - T_{j,j}} \to +\infty$$

Tutti gli altri invece saranno dei numeri reali. Si ottiene quindi che:

$$x = \begin{pmatrix} x_1 \\ \vdots \\ +\infty \\ \vdots \\ x_n \end{pmatrix} \qquad \Rightarrow \qquad \frac{x}{\|x\|} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}$$

Prima di vedere il secondo punto, vediamo il criterio di arresto. Molto semplicemente possiamo prendere:

$$\max_j \frac{|(T_k)_{j,j+1}|}{|(T_k)_{j,j}| + |(T_k)_{j+1,j+1}|} < tol$$

Infatti, se vogliamo che tutti gli elementi sotto la diagonale vadano a zero, possiamo equivalentemente chiedere che il più grande di essi vada a zero.

Vediamo ora il secondo punto, cioè come poter accelerare l'algoritmo. Possiamo pensare di fare qualcosa di simile al metodo delle potenze inverse traslate, cioè prendendo lo shift σ vicino a quel parametro che vogliamo azzerare. Il procedimento è sostanzialmente lo stesso, solo che una volta raggiunta la convergenza, ripetiamo lo stesso procedimento con un altro valore μ. Cioè, dopo la prima convergenza con $\mu = (T_k)_{n,n}$ si passa a $(T_k)_{n-1,n-1}$ e così via.
Abbiamo quindi che l'algoritmo diventa

Terzo Algoritmo di Iterazione QR
Sia $T_0 = \hat{Q}_0^H A \hat{Q}_0$ e $\mu \in \mathbb{C}$
Per $k = 0, 1, ...$
$T_k - \mu I = Q_k R_k$
$T_{k+1} := R_k Q_k + \mu I$

Anche se non ci sono matrici inverse, il metodo corrisponde per filo e per segno con il metodo delle potenze inverse.

Osservazione. All'inizio prendiamo $\mu = (T_0)_{n,n}$ e rimarrà lo stesso finché l'elemento $|(T_k)_{n,n-1}|$ non sarà sufficientemente piccolo. A quel punto si prenderà $\mu = (T_k)_{n-1,n-1}$

Osservazione. Questa procedura lascia lo spettro della matrice invariato. Infatti:

$$T_{k+1} = R_k Q_k + \mu I \overset{\star}{=} Q_k^H (T_k - \mu I) Q_k + \mu I = Q_k^H T_k Q_k - \underbrace{\mu Q_k^H Q_k}_{\mu I} + \mu I = Q_k^H T_k Q_k$$

Dove in $\star$ si è utilizzato che $Q_k^H (T_k - \mu I) = R_k$

Osservazione. La velocità di convergenza segue la stessa logica del metodo delle potenze inverse traslate:

$$\left| \frac{\lambda_{p-1} - \mu}{\lambda_p - \mu} \right|^k$$

Dove abbiamo ordinato gli auto valori in modo decrescente $|\lambda_1 - \mu| \geq |\lambda_2 - \mu| \geq \cdots \geq |\lambda_n - \mu|$

L'utilizzo di questa strategia accelera di molto. Con l'utilizzo di quest'algoritmo così semplice, potrebbero sorgere un sacco di problemi di continuo, per esempio se gli autovalori non sono semplici in modulo o altro ancora, ma nel corso del secolo scorso sono stati fatti nuovi algoritmi per risolvere a questi problemi.
Un esempio, con le matrici si può utilizzare Shur per arrivare ad avere una matrice triangolare a blocchi.

1.3 Analisi di Stabilità

Quando creo U per la decomposizione di Schur e trovo T che approssima R, sono effettivamente quelle di A? Oppure sono almeno vicine?
La risposta è si, infatti il seguente teorema dice:

Teorema 1.3.1

Supponiamo che l'iterazione QR converga dopo $\overline{k}$ iterazioni, allora le matrici $T_{\overline{k}}$ e $U_{\overline{k}}$ ottenute dall'iterazione soddisfano:

$$T_{\overline{k}} = Q^H (A + E) Q \quad \text{con } \|E\|_2 = \mathcal{O}(u\|A\|_2)$$

Con Q matrice unitaria e:

$$U_{\overline{k}}^H U_k = I + E \quad \text{con } \|E\|_2 = \mathcal{O}(u)$$

Con u valore eps della macchina.

Non daremo una vera e propria dimostrazione del teorema, ma faremo delle osservazioni.

Osservazione. La prima affermazione può essere riscritta come: $\exists Q \in \mathbb{C}^{n \times n}$ unitaria tale che $Q^H T_{\overline{k}} Q = A + E$ molto vicino ad A. In maniera analoga, $U_{\overline{k}}$ è molto vicina all'essere una matrice unitaria.
Quindi il risultato che ottengo è molto robusto, cioè una matrice "praticamente" unitaria e una matrice ottenuta dalla decomposizione di Schur di una matrice molto vicina ad A

1.4 Risultati di Perturbazione

Concludiamo l'argomento degli autovalori con un risultato di perturbazione (che ritornerà utile in seguito). Prima di fare ciò, richiamiamo il teorema di Bauer-Fike:

Teorema 1.4.1: Toerema di Bauer-Fike

Per ogni λ autovalore della matrice perturbata, esiste un autovalore $\overline{\lambda}$ della matrice A tale che:

$$|\lambda - \overline{\lambda}| \leq \kappa(X)\|E\|$$

In questo modo, si fa dipendere l'accuratezza della precisione degli autovettori (ne bastano due quasi allineati per rovinare tutto). Il bello di questo risultato è che è generale, però molto debole.

Esiste un altro teorema che serve per tenere conto dei singoli autovalori.

Teorema 1.4.2

Sia $A \in \mathbb{C}^{n\times n}$ e siano (λ, x, y) rispettivamente un autovalore semplice in modulo di A con i rispettivi autovettori destro e sinistro di norma euclidea unitaria. Allora esiste un intorno dell'origine in cui sono definite le funzioni $\lambda(\varepsilon)$ e $x(\varepsilon)$ tali che:

(*i*) $\lambda(0) = \lambda$ e $x(0) = x$

(*ii*) $(A + \varepsilon E)x(\varepsilon) = x(\varepsilon)\lambda(\varepsilon)$ con $E \in \mathbb{C}^{n\times n}$ di norma unitaria e $\lambda(\varepsilon)$ semplice in modulo

(*iii*) $\lambda'(0) = \frac{y^H E x}{y^H x}$ da cui segue che:

$$\lambda(\varepsilon) = \lambda + \varepsilon\frac{y^H E x}{y^H x} + \mathcal{O}(\varepsilon^2) \quad \text{per } \varepsilon \to 0$$

Osservazione. In questo contesto, posso definire $\lambda\varepsilon$ che ci dice come varia λ al variare di A. Notiamo che $\lambda(\varepsilon)$ è proprio autovalore di $A + \varepsilon E$. In questo modo abbiamo sganciato E dalla sua norma e per questo possiamo prendere E con norma 1. Posso allora definire $\lambda(\varepsilon)$ autovalore di $(A + \varepsilon E)$ con autovettore $x(\varepsilon)$.
Queste sono certamente perturbazioni più specifiche, però sono fatte con continuità, in quanto ho:

$$(\lambda(\varepsilon), x(\varepsilon)) \xrightarrow{\varepsilon\to 0} (\lambda, x)$$

Per il terzo punto, invece, c'è uno sviluppo di Taylor che spiega l'andamento di λ al variare di ε (in un intorno di 0). A che cosa ci serve?

$$|\lambda - \lambda(\varepsilon)| = \varepsilon\frac{y^H E x}{y^H x}$$

Qui abbiamo che ε rappresenta $\|E\|$ del teorema di Bauer-Fike, mentre la frazione rappresenta il ruolo di $\kappa(X)$. Detto così però non ci dice granché. Andiamo a svilupparlo:

$$\left|\frac{y^H E x}{y^H x}\right| \leq \frac{\|y\| \cdot \|E\| \cdot \|x\|}{|y^H x|} = \frac{1}{|y^H x|}$$

Quindi l'unica cosa che ci da fastidio è il denominatore

Osservazione. Se A è Hermititana, cioè $A^H = A$, allora segue che $y = x$, da cui segue che:

$$\frac{1}{y^H x} = \frac{1}{\|x\|^2 = 1}$$

Quindi non è amplificato. Una cosa simile si ottiene anche per A normale.

Osservazione. Sia A un blocco di Jordan. In questo caso il teorema non si applica in quanto λ non è semplice in modulo.

Esempio 1. *Supponiamo di avere*

$$A = \begin{pmatrix} \lambda & 1 \\ 0 & \lambda \end{pmatrix}$$

Si vede facilmente che i suoi autovettori destro e sinistro sono:

$$x = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} \lambda & 1 \\ 0 & \lambda \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \lambda \\ 0 \end{pmatrix} = \lambda \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$y = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} \lambda & 1 \\ 0 & \lambda \end{pmatrix} = \begin{pmatrix} 0 & \lambda \end{pmatrix} = \lambda \begin{pmatrix} 0 & 1 \end{pmatrix}$$

Da cui segue subito che $y^H x = 0$, da cui segue subito che:

$$\frac{1}{|y^H x|} = +\infty$$

Osservazione. Risultati del tutto analoghi possono essere ottenuti perturbando anche x

Osservazione. Se A ha autovalore λ con un blocco di Jordan di dimensione al più p, allora vale:

$$|\lambda(A + \varepsilon E) - \lambda(A)| \leq \gamma e^{\frac{1}{p}}$$

2 Definizioni Generali

2.1 Buona Posizione di un problema dato

Supponiamo di avere una funzione f, un dato x e vogliamo calcolare $y = f(x)$.
Supponiamo di avere $\overline{x} \approx x$ e volgiamo capire quanto $\overline{y} = f(\overline{x})$ sia vicino (o lontano) a y

Definizione 2.1.1: Distanza in Senso Assoluto

Diremo che $\overline{y}$ è **vicino** a y **in senso assoluto** se:

$$|\overline{y} - y| \simeq C(x)|\overline{x} - x|$$

Dove $C(x) \in \mathbb{R}$ è detto **Numero di Condizionamento Assoluto**

In prima approssimazione, se f è sufficientemente regolare, allora si ha che:

$$\overline{y} - y = f(\overline{x}) - f(x) = \frac{f(\overline{x}) - f(x)}{\overline{x} - x}(\overline{x} - x) \simeq f'(x)(\overline{x} - x)$$

Da cui segue che $C(x) = |f'(x)|$

Definizione 2.1.2: Distanza in Senso Relativo

Diremo che $\overline{y}$ è **vicino** a y **in senso relativo** se:

$$\frac{|\overline{y} - y|}{|y|} \simeq \kappa(x)\frac{|\overline{x} - x|}{|x|}$$

Dove $\kappa(x) \in \mathbb{R}$ è detto **Numero di Condizionamento Relativo**

In questo caso abbiamo:

$$\frac{|\overline{y} - y|}{|y|} \approx \frac{|f'(x)|}{|y|} \cdot \frac{|\overline{x} - x|}{|x|}|x| = \frac{|f'(x)| \cdot |x|}{|f(x)|}\frac{|\overline{x} - x|}{|x|}$$

Da cui segue che:

$$\kappa(x) \approx \frac{|f'(x)| \cdot |x|}{|f(x)|}$$

Diremo che un problema è **Ben posto** (nel senso che dipende con continuità dai dati) se $C(x)$ è moderato (buona condizione in senso assoluto) o se $\kappa(x)$ è moderato (buona condizione in senso relativo)

Esempio 2. *Sia la funzione $f(x) = \sqrt{x}$, segue immediatamente che:*

$$f'(x) = \frac{1}{2}\frac{1}{\sqrt{x}}$$

Da cui si ottiene che:

$$C(x) = \frac{1}{2}\frac{1}{\sqrt{x}} \qquad e \qquad \kappa(x) = \frac{1}{2}\frac{1}{\sqrt{x}}\frac{|x|}{\sqrt{x}} = \frac{1}{2}$$

Quindi la nostra funzione è ben posta in senso relativo ma è mal posta in senso assoluto

In generale, quando ci sono dei problemi mal posti, si utilizzano dei metodi fatti apposta per questo tipo di problemi. Ma noi in generale vedremo principalmente quelli ben posti.

2.2 Consistenza, Stabilità e Convergenza di un Metodo

Consideriamo un problema ben posto del tipo:

$$F(x, d) = 0$$

dove d rappresenta i dati dati del problema e x rappresenta la sua soluzione (non necessariamente scalare, può essere anche vettore). Questo problema, scritto in questo modo, si dice che è scritto in **Forma Implicita**. Un problema è scritto invece in **Forma Esplicita** se è della forma $x = f(d)$. In generale dovremo abituarci a vedere i problemi scritti in forma implicita, perché nella maggior parte dei casi non ne vedremo una forma esplicita.

Un metodo numerico corrisponde a risolvere il problema dato con una successione del tipo:

$$F_n(x_n, d_n) = 0$$

Ovviamente si cerca una successione tale che $x_n \to x$ per $n \to +\infty$. Sappiamo anche che $F_n \approx F$ e $d_n \approx$ sono delle approssimazioni dei dati iniziali (a volte si prende direttamente gli stessi dati)

Diamo ora delle definizioni formali per questi concetti appena introdotti:

Definizione 2.2.1: Consistenza

Supponendo che il dato d sia ammissibile per F_n, il metodo $F_n(x_n, d_n)$ si dice **consistente** se:

$$F_n(x, d) \xrightarrow{n\to+\infty} 0$$

Cioè se:

$$F_n(x, d) - F(x, d) \xrightarrow{n\to+\infty} 0 \qquad \Leftrightarrow \qquad F_n \xrightarrow{n\to+\infty} 0$$

In parole semplici abbiamo che tende a 0 quando mettiamo la soluzione esatta. *Ovviamente si ha che $F_n(x_n, d_n)$, perché è la soluzione esatta del sistema perturbato*

Definizione 2.2.2: Consistenza Forte

Un metodo si dice fortemente consistente se:

$$F_n(x, d) = 0, \ \forall n \in \mathbb{N}$$

Esempio 3. *Le iterazioni stazionarie sono esempi di metodi iterativi fortemente consistenti.*

$$x_{n+1} = P^{-1}Nx_n + P^{-1}b \quad \Leftrightarrow \quad F_n(x_n, d) = 0$$

Infatti:

$$x^* = P^{-1}Nx^* + P^{-1}b \quad \Rightarrow F_n(x^*, d) = 0$$

Osservazione. Tutti i metodi di punto fisso sono metodi fortemente consistenti, con $x_{n+1} = \phi(x_n)$

Osservazione. Tutti i problemi del tipo $F(x, d) = 0$ con limiti, derivate e integrali non possono essere fortemente consistenti.

Definizione 2.2.3: Stabilità e buona posizione di un problema

Il metodo $F_n(x_n, d_n)$ si dice **Stabile** o **ben posto** se per ogni n fissato abbiamo:

(i) esiste x_n per ogni dato d_n

(ii) la soluzione è unica e le soluzioni sono riproducibili

(iii) x_n dipende con continuità dai dati, cioè:

$$\forall \mu > 0, \exists C_n = C_n(\mu, d_n) : |\delta d_n| < \mu \Rightarrow |\delta x_n| \leq C_n |\delta d_n|$$

Dove si ha che δd_n è la perturbazione relativa a d_n e δx_n relativa a x_n

Notiamo che qui gioca molto il ruolo del numero di condizionamento.

Definizione 2.2.4: Convergenza

Il metodo $F_n(x_n, d_n) = 0$ si dice **convergente** se:

$$\forall \varepsilon > 0, \exists n_0 = n_0(\varepsilon), \exists \delta > 0 : \forall n > n_0(\varepsilon), \forall d_n : |d - d_n| > \delta \Rightarrow |x - x_n| < \varepsilon$$

Dove si ha che $x = x(d)$ e $x_n = x_n(d)$

Questi tre concetti sono strettamente collegati fra loro. Infatti:

Teorema 2.2.5: Equivalenza di Lax-Rightmayer

Sia $F(x, d) = 0$ un problema ben posto e sia $d_n \xrightarrow{n \to +\infty} d$. Sia $F_n(x_n, d_n) = 0$ metodo consistente. Allora il metodo è stabile se e solo se è convergente.

Considerazioni Aggiuntive. *Quello che sostanzialmente dice questo teorema è che se il metodo è consistente, allora convergenza e stabilità sono equivalenti, cioè ne basterà uno per avere entrambi. È un concetto simile alla "robustezza" del metodo*

Dimostrazione. Limitiamo la dimostrazione solo al caso lineare con $Lx - d = 0$ e $L_n x_n - d_n = 0$
Osserviamo che se abbiamo un metodo consistente, allora abbiamo che:

$$F_n(x, d) = L_n x - d \xrightarrow{n \to +\infty} 0$$

Cioè abbiamo che:

$$L_n x - Lx \to 0 \quad \Rightarrow \quad (L_n - L)x \to 0 \quad \Rightarrow \quad L_n \to L$$

Nel caso volessimo lavorare anche con dati perturbati avremmo che (per $n \to +\infty$):

$$L_n x - d_n = L_n x - d + d - d_n = (L_n x - d) + (d - d_n) = (L_n - L)x + (d - d_n) \to 0$$

Mostriamo la prima implicazione, cioè che la stabilità implica la convergenza:

$$x - x_n = L_n^{-1} L_n x - L_n^{-1} L_n x_n + L_n^{-1} Lx - L_n^{-1} Lx = L_n^{-1}(L_n x - Lx) + L_n^{-1}(Lx - L_n x_n)$$

Ora dobbiamo mostrare che tutta questa quantità tende a 0.
Noi però sappiamo che $L_n x_n - d_n = 0$, quindi abbiamo che $x_n = L_n^{-1} d_n$. Sappiamo anche che il metodo è stabile, per cui, per le definizioni che abbiamo dato, abbiamo che $|L_n^{-1}|$ è limitato. *Questo era vero anche perché* $x_n = f(d_n) \Rightarrow f'(d) = L_n^{-1}$
Segue quindi che:

$$|x - x_n| \leq |L_n^{-1}| \cdot |L_n x - Lx| + |L_n^{-1}| \cdot |Lx - L_n x_n|$$

La prima quantità tende a zero per la definizione di consistenza, la seconda tende a zero perché per ipotesi avevamo che $|d - L_n x_n| \to 0$.
Quindi tutto tende a 0

Mostriamo ora l'implicazione opposta, cioè che la convergenza implica la stabilità.
Prendiamo $|x_n(d + \delta d) - x_n(d)|$. Vogliamo vedere quanto questa quantità sia piccola:

$$\begin{aligned} |x_n(d + \delta d) - x_n(d)| &= |x_n(d + \delta d) - x_n(d) + x(d) - x(d) + x(d + \delta d) - x(d + \delta d)| = \\ &= \big|\big(x_n(d + \delta d) - x(d + \delta d)\big) - \big(x_n(d) - x(d)\big) - \big(x(d) - x(d + \delta d)\big)\big| \leq \\ &= |x_n(d + \delta d) - x(d + \delta d)| + |x_n(d) - x(d)| + |x(d) - x(d + \delta d)| \end{aligned}$$

Abbiamo per ipotesi che il metodo è convergente e, poiché sono soluzioni esatte sullo stesso dato, abbiamo che le prime quantità sono rispettivamente minori di ε_1 e ε_2. Abbiamo anche che il metodo è ben posto, quindi per un n sufficientemente grande si ha che la terza quantità è più piccola di $C_1|\delta d|$. Mettendo tutto insieme si ha che la quantità iniziale è più piccola di:

$$|x_n(d + \delta d) - x_n(d)| < \varepsilon_1 + \varepsilon_2 + C_1|\delta d|$$

A questo punto abbiamo sostanzialmente finito, perché per un n sufficientemente grande si ha che:

$$\varepsilon_1, \varepsilon_2 < C_2|\delta d|$$

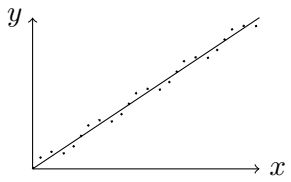
Da cui, mettendo tutto insieme, la quantità iniziale è più piccola di $C_3|\delta d|$. Quindi la soluzione ottenuta dal metodo dipende dai dati □

3 Approssimazione per Interpolazione

3.1 Prime Definizioni

Cerchiamo di capire prima quale sia il problema che puntiamo a risolvere. Sostanzialmente i problemi che risolviamo con l'approssimazione per interpolazione sono due:

1. Date coppie di numeri o di valori $\{(x_i, y_i)\}_{i\in\{1,\dots,n\}}$, vogliamo determinare una funzione $\tilde{f}$ opportuna che approssima il comportamento degli y_i, per esempio mediante **interpolazione**, cioè passando per alcuni dei punti considerati, cioè $y_i = \tilde{f}(x_i)$ per qualche i. Se invece passa per tutte, la funzione si dice **Interpolante**.



   Di solito chiaramente si punta a qualcosa di facile

2. Data una funzione $f : I \to \mathbb{R}$ con dei punti $x_0, ..., x_n \in I$ (possibilmente distinti), voglio trovare una funzione $\tilde{f}$ più facile da gestire tale che:

$$\forall i \in \{0, 1, ..., n\}, \tilde{f}(x_i) = f(x_i$$

Notiamo subito che i due problemi sono equivalenti, in quanto mi basta chiamare $y_i = f(x_i)$.

Perché tutto questo? Perché non tutte le funzioni sono facili da studiare. Prendiamo per esempio:

$$f(x) = \int_0^t \sin^2 te^{\frac{t^2}{2}+2}dt$$

Le funzioni $\tilde{f}$ che stiamo cercando possono essere di vario tipo, per esempio possono essere polinomiali:

$$\tilde{f}(x) = a_0 + a_1x + \cdots + a_nx^n$$

Oppure possono essere circolari:

$$\tilde{f}(x) = a_0 + a_1e^{ix} + \cdots + a_ne^{inx}$$

O addirittura possono razionali:

$$\tilde{f}(x) = \frac{p_x(x)}{q_n(x)}$$

Noi ci limiteremo solo alle interpolazioni complete (cioè di tutti i nodi) di polinomi.

Supponiamo ora $x_i \neq x_j, \forall i, j$ tutti nodi distinti.

Teorema 3.1.1

Date le coppie $\{(x_i, y_i)\}_{i\in\{0,1,\dots,n\}}$ con $x_i \neq x_j$, esiste ed è unico il polinomio di grado n tale che:

$$p_n(x_n) = y_n$$

Questa è la condizione di interpolazione

Dimostrazione. Cominciamo con il dimostrare prima l'unicità.
Supponiamo esista anche $q_n \in \mathbb{P}_n$ tale che $q_n(x_i) = y_i = p_n(x_i)$. Definisco poi:

$$d_n(x) = q_n(x) - p_n(x)$$

polinomio di grado minore uguale a n. Si sa facilmente che $d_n(x_i) = 0$ per ogni i, quindi ha almeno $n+1$ zeri.
Ma questo è un polinomio di grado al massimo n, quindi necessariamente è identicamente nullo, quindi:

$$p_n(x) = q_n(x)$$

Dimostriamone ora l'esistenza.
Vogliamo che sia verificata la condizione di interpolazione di $p_n(x) = a_0 + a_1x + \cdots + a_nx^n$, cioè:

$$\begin{array}{rcl} x_0: & a_0 + a_1x_0 + a_2x_0^2 + \cdots + a_nx_0^2 & = y_0 \\ x_1: & a_0 + a_1x_1 + a_2x_1^2 + \cdots + a_nx_1^2 & = y_1 \\ \vdots & \vdots & \vdots \\ x_n: & a_0 + a_1x_n + a_2x_n^2 + \cdots + a_nx_n^2 & = y_n \end{array}$$

Notiamo però che questo non è altro che un sistema lineare:

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ \vdots \\ y_n \end{pmatrix}$$

Questa matrice prende il nome di matrice di Vandermonde $V_n(x)$
Si vede facilmente che la matrice non è singolare per x_i distinti.
Si può dimostrare poi che il suo determinante è:

$$\det(V_n(x)) = \prod_{0 \le j \le i \le n} (x_i - x_j)$$

□

Osservazione. Il numero di nodi è legato al grado del polinomio. Per trovarlo ci servono $n+1$ condizioni, quindi per creare la matrice ci servono $n+1$ nodi.

Teorema 3.1.2

Posto $cond_\infty(V_n(\underline{x})) = \|V_n(\underline{x})\|_\infty \cdot \|V_n(\underline{x})^{-1}\|_\infty$ e posto:

$$\kappa_{n,\infty} := \inf_{\underline{x} \ge 0} cond_\infty(V_n(x))$$

Allora si ha che:

$$\kappa_{n,\infty} \ge 2^n \quad \text{per } x_i \ge 0, n \ge 2$$

Dove $\underline{x}$ è il vettore con tutti i nodi.

Considerazioni Aggiuntive. *Questo teorema ci da l'idea di quanto tale matrice sia sensibile. Inoltre, sfruttare questa strategia per trovare tale polinomio è una pazzia, in quanto basta $n = 10$ per creare danni. Va sfruttato solo per n piccolo*

Osservazione. Con Matlab abbiamo la funzione `polifit` che permette di determinare il polinomio interpolante (a cui diamo le coppie). Inoltre, se è il caso, ci dice anche se la matrice è ben condizionata oppure mal condizionata. Ci da anche la possibilità di determinare il polinomio interpolante anche se il numero delle coppie che diamo è minore del grado del polinomio

Analizziamo proprio quest'ultimo caso. Sia quindi di avere una matrice $\mathbb{R}^{(m+1)\times(n+1)}$con $m < n$, allora in questo caso abbiamo che la matrice è alta e della forma:

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^m \\ 1 & x_1 & \cdots & x_1^m \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \cdots & x_n^m \end{pmatrix}$$

Da cui segue che il polinomio interpolante è:

$$p_n(x) = a_0 + a_1 x + \cdots + a_m x^m$$

In questo caso, riprendendo la notazione che avevamo usato in precedenza, siamo nel caso di un sistema lineare sovradeterminato $V_n \underline{a} = \underline{y}$ con $\underline{a} \in \mathbb{R}^{m+1}$ e $\underline{y} \in \mathbb{R}^{n+1}$. Quello che si fa è sostanzialmente minimizzare le distanze dei vari punti, cioè trovare $\min \|\underline{y} - V_x \underline{a}\|$, quindi applicare il metodo dei minimi quadrati.

Nel caso in cui invece $m > n$, abbiamo una matrice larga, quindi siamo in un problema sottodeterminato, da cui segue che la soluzione non è unica.

Attenzione. Queste funzioni non passano per tutti i nodi, per questo ci dovranno essere delle approssimazioni. In questo caso non ci sarà interpolazione.

Da qui in avanti studieremo solo il caso completo e studieremo vari modi per arrivare allo stesso polinomio.

3.2 Forma di Lagrange

Definizione 3.2.1: Polinomi in Forma di Lagrange

Siano $x_0, ..., x_n \in I$ nodi distinti. Definiamo il **Polinomio di Lagrange** il polinomio definito come:

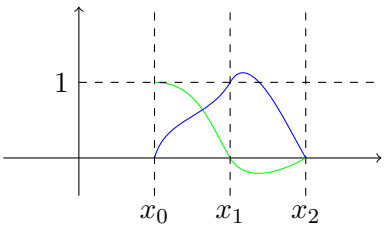
$$L_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j} = \frac{(x - x_0)\cdots(x - x_{i-1})(x - x_{x+1})\cdots(x - x_n)}{(x_i - x_0)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_n)}$$

Questo è un polinomio di grado n.

Osservazione. Questo polinomio assume dei valori speciali nei nodi, infatti:

$$L_i(x_j) = \begin{cases} 1 & \text{se } j = i \quad \text{Qui si semplificano tutti i termini} \\ 0 & \text{se } j \neq i \quad \text{Qui c'è un fattore al numeratore nullo} \end{cases}$$

Questo però non implica che la funzione sia limitata, sappiamo solo che certamente vale $L_i(x_j) = \delta_{i,j}$, con $\delta_{i,j}$ la delta di Kronecker.



Disegnati ci sono dei tratti di polinomi di Lagrange, in verde $L_0(x)$ e in blu $L_1(x)$

Già da qua si può vedere che anche questi non sono particolarmente buoni per il calcolo della funzione interpolante.

Osservazione. I polinomi $\{L_0(x), ..., L_n(x)\}$ rappresentano una base di $\mathbb{P}_n$.
Infatti, se prendiamo una loro combinazione lineare e la poniamo uguale a 0 si ha che:

$$\alpha_0 L_0(x) + \cdots + \alpha_n L_n(x) = 0$$

In particolare, se calcoliamo questa combinazione lineare sui vari nodi i si ha che tutti i termini $L_j(x_i)$ si annullano per come sono stati definiti, quindi:

$$\alpha_0 L_0(x) + \cdots + \alpha_n L_n(x) = 0 = \alpha_i L_i(x_i) = 0$$

Ma sappiamo, sempre in quanto $L_i(x_i)$ è un polinomio di Lagrange, che $L_i(x_i) = 1$, quindi:

$$\alpha_i L_i(x) = \alpha_i = 0$$

Per l'arbitrarietà di i, questo è vero per ogni nodo, da cui segue che tutti gli α_i sono nulli, quindi sono linearmente indipendenti. Possiamo dire che sono una base di $\mathbb{P}_n$ in quanto sono esattamente $n+1$ e la dimensione dello spazio vettoriale $\mathbb{P}_n$ è proprio $n+1$. Da cui segue che:

$$p_n(x) \in \mathbb{P}_n \quad \Rightarrow \quad p_n(x) = \sum_{i=0}^{n} \beta_j L_j(x)$$

Se p_n è il polinomio interpolante, allora si ha che:

$$\forall i \in \{0, ..., n\}, p_n(x_i) = y_i$$

Cioè:

$$p_n(x_i) = \sum_{j=0}^{n} \beta_j L_j(x_i) = \beta_i L_i(x_i) = y_i \quad \Rightarrow \quad \beta_i = y_i$$

Da cui si ottiene definitivamente che:

$$p_n(x) = \sum_{j=0}^{n} y_j L_j(x_i)$$

Definizione 3.2.2: Polinomio Interpolante in forma di Lagrange

Si definisce il **Polinomio Interpolante in forma di Lagrange** il polinomio:

$$p_n(x) = \sum_{j=0}^{n} y_j L_j(x)$$

Vediamo un primo esercizio, tipico dell'esame:

Esercizio. Siano $x = [0, 1, 3]$ e $y = [1, 3, 2]$ due vettori, vogliamo trovare la funzione interpolante di queste coppie di numeri e stimare con il polinomio interpolante i possibili valori di $xy = 2$

Soluzione. Andiamo a calcolare quanto valgono i polinomi in forma di Lagrange:

$$x_0 : L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \frac{(x - 1)(x - 3)}{(-1)(-3)} = \frac{1}{3}(x - 1)(x - 3)$$

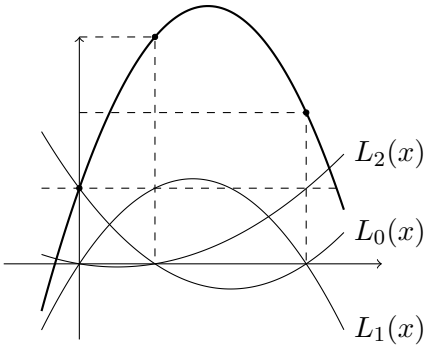
$$x_1 : L_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} = \frac{x(x - 3)}{(1)(-2)} = -\frac{1}{2}x(x - 3)$$

$$x_2 : L_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} = \frac{x(x - 1)}{(3)(2)} = \frac{1}{6}x(x - 1)$$

Da cui segue che:

$$p_2(x) = L_0(x) + 3L_1(x) + 2L_2(x) = -\frac{7}{6}x^2 + \frac{17}{6}x + 1$$

Quello che otterremo sarà:



Per capire invece indicativamente i possibili valori di $xy = 2$, ci basta:

$$p_2(2) = L_0(2) + 3L_1(2) + 2L_2(2) = \frac{10}{3}$$

■

3.3 Analisi dell'Errore

Supponiamo di avere una famiglia crescente di nodi, cioè di avere $\{x_0^{(0)}\}$, $\{x_0^{(1)}, x_1^{(1)}\}$,... tutti contenuti in $[a, b]$. Diremo che c'è convergenza se con $n \to +\infty$. Nel dettaglio: data una funzione $f : [a, b] \to \mathbb{R}$, il polinomio di Lagrange nei nodi delle famiglie dei nodi converge se:

$$p_n(x) \xrightarrow{n \to +\infty} f(x) \text{ uniformemente in } x \in [a, b]$$

Affinché questa cosa avvenga, dobbiamo chiarire due questioni:

- Fare la stima dell'errore con i nodi (che in un certo senso è quanto abbiamo già visto)
- Studiare la convergenza all'aumentare dei nodi

Con il prossimo teorema, potremo chiarire parte della seconda questione:

Teorema 3.3.1

Siano $x_0, x_1, ..., x_n$ punti distinti e sia $x^* \in [a,b]$ e $[a,b]$ intervallo contenente tutti i punti $\{x_i\}$. Supponiamo di avere $f \in C^{n+1}([a,b])$. Allora esiste $\xi \in]a,b[$ tale che:

$$E_n(x^*) := f(x^*) - p_n(x^*) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega(X^*)$$

Dove si ha che $\xi = \xi(x^*)$, p_n è il polinomio interpolante di f nei nodi $x_0, ..., x_n$ e $\omega(x) = (x - x_0)\cdots(x - x_n)$, detto **Polinomio Normale**, di grado n

Considerazioni Aggiuntive. *Si può vedere come in questa dimostrazione, l'utilizzo dell'analisi è molto evidente.*
Qui l'errore è definito come $f(x^*) - p_n(x^*)$ ed è puntuale, cioè dipende punto per punto, quindi si ha che necessariamente ξ dipende dalla scelta del punto x^*. Inoltre, il fatto che ci sia al denominatore un $(n+1)!$ ci dice che l'errore tende ad essere molto piccolo, derivate e $\omega(x)$ permettendo, in quanto non possiamo sapere a priori che valori assumono. Quello che possiamo fare però è poterlo stimare con le norme infinite.

Dimostrazione. Supponiamo di avere i punti distinti $x_i \neq x_j, \forall i,j$ e supponiamo anche $x^* \neq x_j, \forall j$. (Altrimenti si avrebbe banalmente che $E_n(x^*) = 0$, perché è proprio interpolante) Definiamo poi $F(x)$ come:

$$F(x) := f(x) - p_n(x) - \frac{f(x^*) - p_n(x^*)}{\omega(x^*)}\omega(x)$$

Notiamo subito che $F \in C^{n+1}([a,b])$, in quanto è somma, prodotto e composizione di funzioni di classe C^{n+1}. Andiamo a vedere che valore assume F nei vari punti.
Sui nodi abbiamo che:

$$F(x_i) = \underbrace{f(x_i) - p_n(x_i)}_{0} - \frac{f(x^*) - p_n(x^*)}{\omega(x^*)}\underbrace{\omega(x^*)}_{0} = 0 \qquad \forall i$$

Invece su x^* si ha che:

$$F(x^*) = f(x^*) - p_n(x^*) - \frac{f(x^*) - p_n(x^*)}{\omega(x^*)}\omega^* = 0$$

Ne segue quindi che F ha almeno $n+2$ zeri. Per il teorema di Rolle applicato su F, si ha che F' ha almeno $n+1$ zeri. Riapplicandolo nuovamente, F'' ha almeno n zeri. Andando avanti così si ottiene che $F^{(n+1)}$ ha almeno uno zero. Chiamiamo ξ lo zero di $F^{(n+1)}$, cioè $F^{(n+1)}(\xi) = 0$
Ne segue quindi che:

$$F^{(n+1)}(x) = f^{(n+1)}(x) - \underbrace{p_n^{(n+1)}(x)}_{0} - \frac{f(x^*) - p_n(x^*)}{\omega(x^*)}\underbrace{\omega^{(n+1)}(x)}_{0}$$

Da cui:

$$f^{(n+1)}(\xi) = \frac{f(x^*) - p_n(x^*)}{\omega(x^*)}(n+1)! \quad \Rightarrow \quad f(x^*) - p_n(x^*) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega(x^*)$$

□

Osservazione. Utilizzando il fatto che $\|f\|_\infty = \max|f(x)|$, con le norme e i valori assoluti abbiamo che:

$$|E_n(x^*)| \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}\omega(x^*) \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}\|\omega\|_\infty$$

Osservazione. In generale, sapendo che $\|\omega\|_\infty \leq |b-a|^{n+1}$, direttamente da $|x - x_i| \leq |b-a|$, si ha convergenza se:

$$\lim_{n\to+\infty} \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}(b-a)^{n+1} = 0$$

Stiamo confrontando $\|f^{(n+1)}\|$ e $\frac{(b-a)^{n+1}}{(n+1)!}$ e sappiamo che il secondo tende a zero. Del primo a priori non lo sappiamo. Questa stima è molto grossolana

Esempio 4. *Sia $f(x) = e^x$ con $x \in [a,b]$ e siano $x_0, x_1, ..., x_n$ nodi distinti. Allora sappiamo che:*

$$\|f^{(n+1)}\|_\infty = e^b$$

Questo perché la funzione è strettamente crescente e $[a,b]$ è chiuso e limitato. Da questo segue che:

$$\|E\|_\infty = \max_{x\in[a,b]}|E(x)| \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}(b-a)^{n+1} = e^b\frac{(b-a)^{n+1}}{(n+1)!} \xrightarrow{n\to+\infty} 0$$

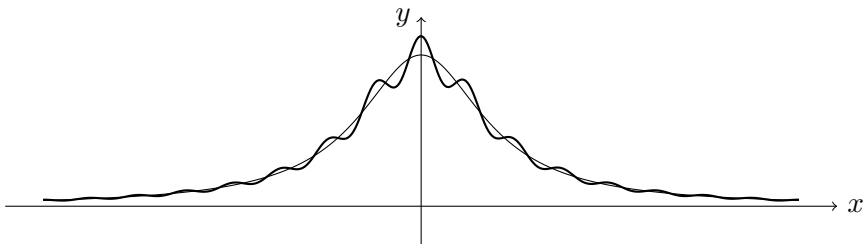
In questo caso si dice c'è compattezza uniforme in $[a,b]$

In generale non va sempre bene, ecco uno degli esempi più famosi.

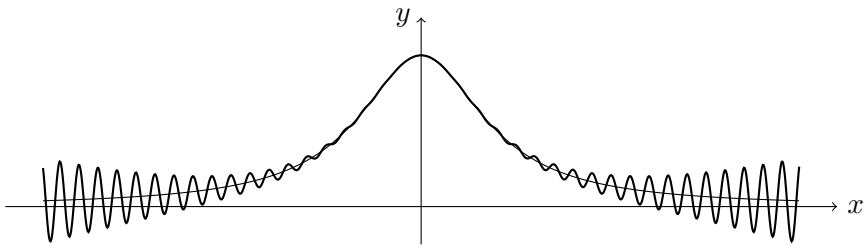
Esempio 5 (Runge)**.** *Sia la funzione:*

$$f : [-5,5] \to \mathbb{R} \qquad f(x) = \frac{1}{1+x^2}$$

Immaginiamo di volerla interpolare, partendo da pochi punti si ha che:



Più aumenta più notiamo che l'interpolazione tende ad allontanarsi a quella che effettivamente è la curva



Continuando ad interpolare, si avranno sempre dei punti che in cui la funzione interpolante esplode. Volendo è possibile dimostrare che vale:

$$\lim_{n\to+\infty} |f(x) - p_n(x)| = \begin{cases} 0 & se\ |x| < 3,63 \\ +\infty & se\ |x| > 3,63 \end{cases}$$

Volendo ci sarebbe un modo per interpolare questa funzione, ma bisogna utilizzare i polinomi ortogonali, cosa che non faremo in questo corso.
*Perché questo succede? Se la funzione fosse definita sul piano complesso avremmo che la funzione non è definita su $z = \pm i$. In questo caso z prende il nome di **polo**. Qui chiaramente non possiamo studiare tutto, ma non possiamo neanche restringerci al dominio $[-5, 5]$ perché in tal caso prenderemmo una palla centrata nell'origine di raggio 5, che appunto comprende i punti in cui la funzione non è definita.*

3.4 Nodi non Equidistanti

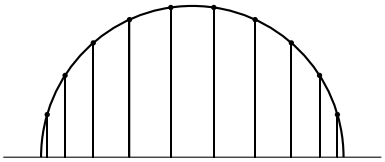
Introduciamo questa sezione perché risultano molto importanti per la risoluzione di alcuni problemi

3.5 Nodi di Chebyshev

I nodi di Chebyshev consistono nel prendere una semicirconferenza e di prendere i punti equidistanti tra di loro sulla circonferenza, per poi proiettarli sull'asse delle ascisse:

$$\hat{x}_k = \cos\left(\frac{2k+1}{2k+2}\pi\right), \quad k \in \{0, ..., n\}$$

Graficamente abbiamo che:



Se aumento il numero dei nodi, non li ho equispaziati su $[a, b]$ ma sull'arco di circonferenza con estremi 1 e -1. A questo punto mi basta fare la trasformazione:

$$[-1, 1] \to [a, b] \qquad \Rightarrow \qquad x_k = \frac{a+b}{2} + \frac{b-a}{2}\hat{x}_k$$

Queste che abbiamo appena ottenuto sono le radici del polinomio di Chebyshev di grado $n + 1$.

Definizione 3.5.1: Polinomi di Chebyshev

I **Polinomi di Chebyshev** sono delle funzioni speciali definite come:

$$T_n(\cos\theta) = \cos(n\theta) \qquad \theta \in [0, 2\pi]$$

Dove n è il grado del polinomio. Se poniamo poi $x = \cos\theta$ allora abbiamo che:

$$T_n(x) = \cos(n \arccos(x)) \qquad x \in [-1, 1]$$

Andiamo a vedere le proprietà di questi polinomi:

- $|T_n(x)| \leq 1$, in quanto è un coseno
- Vediamo adesso dei casi al variare di n:

$$\begin{aligned} n=0: &\quad T_0(x) = 1 \\ n=1: &\quad T_1(x) = \cos\theta = x \\ n=2: &\quad T_2(x) = \cos(2\theta) = 2\cos^2\theta - 1 = 2\underbrace{\cos\theta}_{x}\underbrace{\cos\theta}_{T_1(x)} - \underbrace{1}_{T_0(x)} = 2xT_1(x) - T_0(x) \end{aligned}$$

  Notiamo che abbiamo trovato una ricorrenza (che può essere dimostrata per induzione) tale che:

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \forall n \in \mathbb{N}$$

- Il coefficiente direttore è 2^{n-1} e il rispettivo polinomio monico è:

$$\mathring{T}_n(x) := \frac{1}{2^{n-1}} T_n(x)$$

- Sapendo che $T_n(x) = \cos(n \arccos(x))$, gli zeri di questo polinomio sono:

$$\begin{aligned} T_n(x) = \cos(n\arccos(x)) = 0 \Rightarrow & \\ \Rightarrow n\arccos(x) = (2k-1)\frac{\pi}{2} & \qquad \text{con } k = 1, 2, ..., n \\ \Rightarrow \arccos(x) = \frac{2k-1}{2n}\pi & \qquad \text{con } k = 1, 2, ..., n \\ \Rightarrow x = \cos\left(\frac{2k-1}{2n}\pi\right) & \qquad \text{con } k = 1, 2, ..., n \end{aligned}$$

  Notiamo che se scaliamo a $T_{n+1}(x)$ otteniamo che gli zeri sono:

$$\hat{x}_k = \cos\left(\frac{2k+1}{2n+2}\pi\right)$$

  Che è esattamente quanto avevamo detto precedentemente.

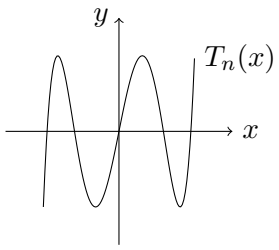
- Andiamo a capire quali sono i punti z_k in cui la funzione assume valori massimi e valori minimi, cioè i punti z_k tali che:

$$T_n(z_k) = \pm 1$$

  Proseguendo in maniera del tutto analoga a quanto fatto precedente (con la sola eccezione che $k = 0, 1, ..., n$) si ha che:

$$z_k = \cos\left(\frac{k}{n}\pi\right) \qquad k = 0, 1, ..., n$$

Osservazione. È fondamentale ricordarsi la ricorrenza che abbiamo trovato nel secondo punto, perché è quello che ci permette di dire che tutti i polinomi di Chebyshev sono effettivamente dei polinomi



Il grafico di $T_n(x)$ con $n = 5$

In Matlab possiamo definire questa funzione utilizzando gli "handle" @, attraverso la sintassi:

$$\texttt{f} = @(\texttt{x},\texttt{n})(\texttt{cos}(5 * \texttt{acos}(\texttt{x})))$$

In questo modo abbiamo definito una funzione a due variabili che calcola il polinomio di Chebyshev di grado n in x.

Teorema 3.5.2

È verificata:

$$\forall \mathring{p} \in \mathbb{P}_n, \text{monico}, \|\mathring{p}\|_{\infty,[-1,1]} \geq \|\mathring{T}\|_{\infty,[-1,1]} = \frac{1}{2^{n-1}}$$

O, equivalentemente:

$$\min_{\mathring{p}\in\mathbb{P}_n} \max_{x\in[-1,1]} |\mathring{p}(x)| \geq \frac{1}{2^{n-1}} = \max_{x\in[-1,1]} |\mathring{T}(x)|$$

Considerazioni Aggiuntive. *Quello che il teorema sostanzialmente ci sta dicendo è che tra tutti i polinomi monici di grado n, il polinomio di Chebyshev è quello con minima norma infinita e il suo valore massimo è pari a* $\frac{1}{2^{n-1}}$

Dimostrazione. Dimostriamolo per assurdo.
Supponiamo per assurdo che esista un polinomio $\mathring{p}_n(x) \in \mathbb{P}_n$ tale che:

$$|\mathring{p}_n(x)| < \frac{1}{2^{n-1}}$$

E di conseguenza anche il suo massimo è minore di $\frac{1}{2^{n-1}}$. Definiamo d_n come:

$$d_n(x) = \mathring{T}_n(x) - \mathring{p}_n(x)$$

Sappiamo che necessariamente $\deg(d_n) = n - 1$, in quanto i coefficienti direttori di $\mathring{T}_n$ e $\mathring{p}_n$ è 1, quindi si cancellano. Nei punti z_k, che ricordiamo essere definiti come:

$$z_k = \cos\left(\frac{k}{n}\pi\right)$$

Abbiamo che:

$$T_n(z_k) = \pm 1 \qquad \Rightarrow \qquad \mathring{T}_n(z_k) = \pm\frac{1}{2^{n-1}}$$

Sapendo questa cosa, possiamo dire con tranquillità che:

$$d_n(z_0) = \frac{1}{2^{n-1}} - \mathring{p}_n(z_0) > 0 \qquad d_n(z_1) = -\frac{1}{2^{n-1}} - \mathring{p}_n(z_1) < 0 \qquad \ldots$$

In generale avremo che

$$d_n(z_{2k}) = \frac{1}{2^{n-1}} - \mathring{p}_n(z_{2k}) > 0 \qquad d_n(z_{2k+1}) = -\frac{1}{2^{n-1}} - \mathring{p}_n(z_{2k+1}) < 0$$

Ci sono quindi n cambi di segno, questo significa che ci sono n zeri (il polinomio è continuo; in particolare lo è quello di Chebyshev per la ricorrenza che avevamo trovato in precedenza). Però il grado del polinomio è $n-1$. Deve seguire necessariamente che d_n è identicamente nullo.
Abbiamo trovato una contraddizione al fatto che tale $\mathring{p}_n$ esista, da cui segue che:

$$|\mathring{p}_n(x)| \geq \frac{1}{2^{n-1}}, \forall x \in [-1,1], \forall \mathring{p}_n \in \mathbb{P}_n \text{ monici}$$

□

Che cosa ce ne facciamo?
Tutto questo ci era servito per l'interpolazione. In particolare, riprendiamo l'errore di interpolazione prendendo $[a,b] = [0,1]$. Avevamo che l'errore era definito come:

$$|E(x^*)| \leq \left\| \frac{f^{(n+1)}}{(n+1)!} \right\|_\infty \cdot \|\omega\|_\infty$$

+ Dove ω era il polinomio nodale, cioè:

$$\omega = \prod_{j=0}^{n} (x - x_j)$$

Allora abbiamo che ω è un polinomio monico di grado $\deg(\omega) = n+1$

Se io volessi minimizzare l'errore il più possibile, ho due scelte:

- Aumentare il grado di ω
- Diminuire $\|\omega\|_\infty$

È proprio a quest'ultimo punto che ci servono i polinomi di Chebyshev.
Infatti, sfruttando il fatto che ω è monico, possiamo prendere:

$$\omega \equiv \mathring{T}_{n+1} \qquad \Leftrightarrow \qquad \mathring{T}_{n+1} = \arg\min_{\mathring{p}_{n+1}}$$

In questo modo abbiamo che:

$$\|\omega\|_\infty = \|\mathring{T}_{n+1}\|_\infty = \frac{1}{2^n}$$

Quindi, se ho grado n, ho una norma piccolissima dell'errore, infatti:

$$|E(x^*)| \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1!)2^n}$$

Però tutto questo è fatto nell'intervallo $[-1,1]$. *Come possiamo generalizzarlo?*
Possiamo fare come avevamo già fatto in precedenza, cioè, se $\hat{x}_j$ sono i nodi, allora:

$$x_j = \frac{a+b}{2} - \frac{b-a}{2}\hat{x}_j$$

Ma la cosa è valida per un qualsiasi punto $x \in [a,b]$, infatti $\exists \hat{x} \in [-1,1]$ tale che:

$$x = \frac{a+b}{2} - \frac{b-a}{2}\hat{x}$$

In questo modo, possiamo riscrivere ω come:

$$\omega(x) = \prod_{j=0}^{n}(x - x_j) = \prod_{j=0}^{n}\left(\frac{b-a}{2}\right)(\hat{x} - \hat{x}_j) = \left(\frac{b-a}{2}\right)^{n+1}\prod_{j=0}^{n}(\hat{x} - \hat{x}_j)$$

In questo modo abbiamo che:

$$\|\omega\|_{\infty,[a,b]} = \left(\frac{b-a}{2}\right)^{n+1}\left\|\prod_{j=0}^{n}(\hat{x} - \hat{x}_j)\right\|_{\infty,[-1,1]} = \left(\frac{b-a}{2}\right)^{n+1}\cdot\frac{1}{2^n} = \frac{(b-a)^{n+1}}{2^{2n+1}}$$

Osservazione. In generale avevamo che $\|\omega\|_{\infty,[a,b]} \leq (b-a)^{n+1}$. Utilizzando i polinomi di Chebyshev abbiamo che tale stima si riduce drasticamente a:

$$|E(x^*)| \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}\cdot\frac{(b-a)^{n+1}}{2^{2n+1}}$$

Teorema 3.5.3: di Bernstein

Supponiamo $f \in C^1([a,b])$. Allora il polinomio p_n interpolante f nei nodi di Chebyshev $x_0, x_1, ..., x_n$ converge uniformemente su $[a,b]$ per $n \to +\infty$

Considerazioni Aggiuntive. *Il fatto che la funzione converga uniformemente implica che tutti i punti tendono a $f(x)$. Inoltre più grande sarà il grado del polinomio, migliore sarà la convergenza*

Esempio 6. *Riprendiamo l'esempio 5 della funzione di Runge. Se utilizziamo i polinomi di Chebyshev ho che l'errore va a 0 per $n \to +\infty$. Infatti:*

Grado del Polinomio	5	10	20	40
$\|\mathbf{E}\|_{\infty,[-\mathbf{5},\mathbf{5}]}$	0,5	0,089	0,015	0,00028

Esercizio (Prova 3/9/2019)**.** È data la funzione $f : [a,b] \to \mathbb{R}$ con $f > 0$ e siano:

$$m_0 = \min_{[a,b]}|f(x)| \qquad \text{e} \qquad M_k = \max_{[a,b]}|f^{(k)}(x)| \quad k = 0, 1, ...$$

1. Sia p_n il polinomio interpolante con nodi di Chebyshev in $[a,b]$. Si stimi:

$$r_k = \max_{[a,b]}\frac{|f(x) - p_{n-1}(x)|}{|f(x)|} \quad \text{cioè l'errore relativo}$$

2. Per $f(x) = \log(x)$ e per $[a,b] = [e, e^2]$, si determini β tale che $r_n = \alpha\beta^n$ con $\alpha = \alpha(n)$ in modo moderato (*che non scoppi*)

Soluzione. 1 Sia $x \in [a,b]$, allora abbiamo che:

$$\frac{|f(x) - p_{n-1}(x)|}{f(x)} \leq \frac{\|f^{(n)}\|_\infty}{n! \cdot m_0} \cdot \|\omega\|_\infty$$

Possiamo dire altro? Sappiamo che:

$$\|\omega\|_\infty = \left(\frac{b-a}{2}\right)^n \cdot \frac{1}{2^{n-1}}$$

Da cui segue che:

$$\frac{|f(x) - p_{n-1}(x)|}{f(x)} \leq \frac{\|f^{(n)}\|_\infty}{n! \cdot m_0} \cdot \|\omega\|_\infty = \frac{\|f^{(n)}\|_\infty}{n! \cdot m_0} \cdot \frac{(b-a)^n}{2^{2n-1}}$$

2 Siano quindi $f(x) = \log x$ e $[a,b] = [e, e^2]$, allora si ha che:

$$f'(x) = \frac{1}{x} \quad f''(x) = -\frac{1}{x^2} \quad f^{(3)}(x) = \frac{2}{x^3} \quad f^{(4)} = -\frac{6}{x^4}$$

Notiamo allora che, per ogni n, si ha che:

$$r_n \leq \frac{\|f^{(n)}\|_\infty}{n! \cdot m_0} \cdot \frac{(b-a)^n}{2^{2n-1}}$$

Notiamo però che si ha che:

$$|f^{(n)}| = \frac{(n-1)!}{x^n} \quad \Rightarrow \quad \|f^{(n)}(x)\|_{\infty,[a,b]} \leq \frac{(n-1)!}{e^n}$$

Cioè la maggiorazione non dipende da x. Avevamo che:

$$m_0 = \min_{[e,e^2]} |f(x)| = |\log e| = 1 \qquad b - a = e(e-1)$$

Andando a sostituire si ha che:

$$r_n \leq \frac{(n-1)!}{e^n} \cdot \frac{1}{n!} \cdot \frac{1}{1} \cdot \frac{e^n(e-1)^n}{2^{2n-1}} = \underbrace{\frac{2}{n}}_{\alpha} \cdot \underbrace{\left(\frac{e-1}{4}\right)^n}_{0 \leq \beta \leq 1} = \alpha\beta^n$$

Notiamo poi che $\alpha \to 0$ per $n \to \infty$, quindi abbiamo finito ■

3.6 Analisi di Stabilità

Supponiamo di avere le coppie (x_i, y_i) con $i \in \{0, ..., n\}$, coppie di osservazioni (y_i può essere $f(x_i)$ per una qualche f, ma non è strettamente necessario). Supponiamo di non conoscere y_i in maniera esatta, ma di conoscere $\tilde{y}_i$ sua perturbazione. Avremo, utilizzando i polinomi in base di Lagrange, allora che:

$$\tilde{p}_n(x) = \sum_{i=0}^{n} \tilde{y}_i L_i(x)$$

In particolare, andando a fare la differenza tra i due polinomi:

$$p_n(x) - \tilde{p}_n(x) = \sum_{i=0}^{n} y_i L_i(x) - \sum_{i=0}^{n} \tilde{y}_i L_i(x) = \sum_{i=0}^{n} (y_i - \tilde{y}_i) L_i(x)$$

In particolare, andandone a prendere prima i valori assoluti e poi le norme infinito, avremo che:

$$|p_n(x) - \tilde{p}_n(x)| \leq \sum_{i=0}^{n} |y_i - \tilde{y}_i| \cdot |L_i(x)| \leq \max_{i\in\{0,\dots,n\}} |y_i - \tilde{y}_i| \sum_{i=1}^{n} L_i(x)$$

$$\|p_n(x) - \tilde{p}_n(x)\|_\infty = \max_{x\in[a,b]} |p_n(x) - \tilde{p}_n(x)| \leq \max_{i\in\{0,\dots,n\}} |y_i - \tilde{y}_i| \cdot \max_{x\in[a,b]} \sum_{i=0}^{n} |L_i(x)|$$

In questo modo abbiamo che la massima perturbazione aumenta con il massimo di $|y - \tilde{y}|$ e da:

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

Questa costante prende il nome di **Costante di Lebesgue**.

Definizione 3.6.1: Costante di Lebesgue

Data una serie di coppie (x_i, y_i) e $(\tilde{y}_i)$ le rispettive perturbazioni, con $x \in [a, b]$, si definisce la **Costante di Lebesgue** la costante:

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

Questo rappresenta il fattore di amplificazione dell'errore / della perturbazione

In sintesi si ha che:

$$\|p_n - \tilde{p}_n\|_\infty \leq \max_{x\in\{0,\dots,n\}} |y_i - \tilde{y}_i| \Lambda_n$$

Osservazione. Facciamo diverse osservazioni:

- Λ_n dipende <u>solo</u> dai nodi scelti $\{x_0, ..., x_n\}$ e non da $f(x_i) = y_i$
- Si può dimostrare che Λ_n, a seconda dei nodi scelti, ha il seguente comportamento:

$$\begin{cases} \Lambda_n \approx \frac{2^n}{n \log n} & \text{Se i nodi sono equivalenti} \\ \Lambda_n \approx \log n & \text{Se si usano i nodi di Chebyshev} \end{cases}$$

- Una n grande può causare seri problemi

Oltre a tutto questo Λ_n è coinvolto anche in un risultato più qualitativo:

Teorema 3.6.2

Sia $f \in C^0([a, b])$, p_n il polinomio di interpolazione relativo a $\{x_0, ..., x_n\}$, allora:

$$\|f - p_n\|_{\infty,[a,b]} \leq (1 + \Lambda_n) \inf_{q\in\mathbb{P}_n} \|f - q\|_{\infty,[a,b]}$$

Considerazioni Aggiuntive. *Quello che sostanzialmente il teorema ci sta dicendo è che la distanza del polinomio interpolante dal migliore polinomio q cresce di un valore pari a $1+\Lambda_n$ (per questo siamo sicuri che cresca, in quanto questo valore è sempre positivo). Sottolineiamo che q nell'enunciato del teorema, non necessariamente è un polinomio interpolante, ma è il polinomio che meglio riduce la distanza di f dal polinomio*

Dimostrazione. Sia $q_n \in \mathbb{P}_n$. Allora si ha che:

$$f(x) - p_n(x) = f(x) - q_n(x) + q_n(x) - p_n(x)$$

In particolare si ha che:

$$|f(x) - p_n(x)| \leq |f(x) - q_n(x)| + |q_n(x) - p_n(x)|$$

Sapendo che $q_n \in \mathbb{P}_n$, possiamo ancora interpolarlo e ottenere che:

$$\begin{aligned}
|q_n(x) - p_n(x)| &= \left| \sum_{i=0}^{n} q_n(x_i) L_i(x) - \sum_{i=0}^{n} f(x_i) L_i(x) \right| \leq \sum_{i=0}^{n} |q_n(x_i) - f(x_i)| \cdot |L_i(x)| \\
&\leq \max_{i \in \{0,\dots,n\}} |q_n(x_i) - f(x_i)| \sum_{i=0}^{n} |L_i(x)| \leq \max_{x \in \{a,b\}} |q_n(x) - f(x)| \sum |L_i(x)| \\
&= \|f - q_n\|_{\infty,[a,b]} \sum_{i=0}^{n} |L_i(x)|
\end{aligned}$$

Dalla disuguaglianza ad inizio dimostrazione abbiamo che:

$$\begin{aligned}
\|f - p_n\|_\infty &\leq \max_{x \in [a,b]} |f(x) - q_n(x)| + \max_{x \in [a,b]} |q_n(x) - p_n(x)| \\
&= \|f - q_n\|_\infty + \|f - q_n\|_\infty \max_{x \in [a,b]} \sum_{i=0}^{n} |L_i(x)| \\
&= \|f - q_n\|_\infty + \|f - q_n\| \Lambda_n = (1 + \Lambda_n) \|f - q_n\|_\infty
\end{aligned}$$

Per l'arbitrarietà di q_n, si ha che vale $\forall q \in \mathbb{P}_n$, quindi anche per l'inf, da cui segue la tesi □

Esercizio. Sia $f(x) = \cos(\pi x) e^{2x}$ con $x \in [-\frac{1}{4}, \frac{1}{4}]$.

1. Determinare il polinomio interpolante a f in $x_0 = -\frac{1}{4}$, $x_1 = 0$, $x_2 = \frac{1}{4}$
2. Supponendo $\tilde{y}_2 \approx 1,63$ (con $y_2 \approx 1,59$), stimare $\|p_2 - \tilde{p}_2\|_\infty$

Soluzione. 1 Per il primo punto possiamo fare come in uno degli esercizi/esempi precedenti.

$$\begin{aligned}
L_0(x) &= \frac{(x-0)(x-\frac{1}{4})}{(-\frac{1}{4}-0)(-\bar{1}4-\frac{1}{4})} = 8x\left(1 - \frac{1}{4}\right) = 8x^2 - 2x \quad f(x_0) = \frac{\sqrt{2}}{2} e^{-\frac{1}{2}} \\
L_1(x) &= \frac{(x-\frac{1}{4})(x+\frac{1}{4})}{(0-\frac{1}{4})(0+\frac{1}{4})} = -16\left(x^2 - \frac{1}{16}\right) = -16x^2 + 1 \quad f(x_1) = 1 \\
L_2(x) &= \frac{(x+\frac{1}{4})x}{(\frac{1}{4}+\frac{1}{4})(\frac{1}{4})} = 8x^2 + 2x \qquad\qquad f(x_2) = \frac{\sqrt{2}}{2} e^{\frac{1}{2}}
\end{aligned}$$

Da questo segue che:

$$p_2(x) = \frac{\sqrt{2}}{2} e^{-\frac{1}{2}} \left(8x^2 - 2x\right) - (16x^2 - 1) + \frac{\sqrt{2}}{2} e^{\frac{1}{2}} (8^2 + 2)$$

2 Dobbiamo cercare:

$$\max_{x\in[a,b]} |p_2(x) - \tilde{p}_2(x)| \qquad \text{con } \tilde{p}_2(x) = y_0 L_0(x) + y_1 L_1(x) + \tilde{y}_2 L_2(x)$$

Infatti:

$$\|p_2(x) - \tilde{p}_2(x)\|_\infty \leq \max_{i\in\{0,1,2\}} |y - \tilde{y}_i| \Lambda_n(x) = (y_2 - \tilde{y}_2)\Lambda_n(x) \approx 0,4\Lambda_n$$

Con Λ_n che ha una stima pessimistica. Otteniamo quindi che:

$$p_2(x) - \tilde{p}_2(x) = 0 + 0 + (y_2 - \tilde{y}_2)L_2(x) = (y_2 - \tilde{y}_2)L_2(x)$$

In particolare otteniamo che:

$$|p_2(x) - \tilde{p}_2| = |y_2 - \tilde{y}_2| \cdot |L_2(x)| \quad \Rightarrow \quad \|p_2 - \tilde{p}_2\|_\infty = |y_2 - \tilde{y}_2| \|L_2\|_\infty = |y_2 - \tilde{y}_2| \approx 0,04$$

Abbiamo che $\|L_2\|_\infty = 1$ in quanto se $L_2(x) = 8x^2 + 2x$, allora $L_2'(x) = 16x + 2$. Otteniamo un punto critico con $x = -\frac{1}{8}$, ed è un punto di minimo. Nell'intervallo $[-\frac{1}{4}, \frac{1}{4}]$, abbiamo che $\frac{1}{4}$ è punto di massimo, quindi:

$$L_2\left(\frac{1}{4}\right) = \frac{1}{2} + \frac{1}{2} = 1$$

Quindi tutto torna ■

Esercizio. Siano $[a,b] = [-1,1]$, $x_0 = -\alpha, x_1 = 0, x_2 = \alpha$ e consideriamo p_2 associato a $\{x_0, x_1, x_2\}$

1. Maggiora l'errore di interpolazione in modo indipendente da x (+ accumulato possibile) per $\alpha = 1$ e $f(x) = \sin x$
2. Dimostra che al variare di $\alpha \in [0,1]$, la migliore stima (ossia il valore di $\|\omega\|_\infty$, del polinomio nodale) sia per $\alpha = \frac{\sqrt{3}}{2}$

Soluzione. 1 Per quanto fatto in precedenza, sappiamo che:

$$\|f - p_1\|_\infty \leq \frac{\|f^{(3)}\|_\infty}{3!}\|\omega\|_\infty = \frac{\|-\cos x\|_\infty}{6}\|\omega\|_\infty = \frac{1}{6}\|\omega\|_\infty$$

Sappiamo poi che:

$$\omega x = (x - x_0)(x - x_1)(x - x_2) = (x + \alpha)(x)(x - \alpha) = x(x^2 - \alpha^2) = x^3 - \alpha^2 x$$

In particolare il polinomio nodale è nullo se e solo se $x = \pm\frac{\alpha}{\sqrt{3}}$. Abbiamo poi che:

$$\omega\left(\pm\frac{\alpha}{\sqrt{3}}\right) = \pm\frac{\alpha}{\sqrt{3}} = \pm\frac{\alpha}{\sqrt{3}}\left(\frac{\alpha^2}{3} - \alpha^2\right) = \sqrt{\alpha}\sqrt{3}\left(-\frac{2}{3}\alpha^2\right) = \mp\frac{2}{3\sqrt{3}}\alpha^3$$

Abbiamo quindi che il suo valore assoluto è:

$$\left|\omega\left(\pm\frac{\alpha}{\sqrt{3}}\right)\right| = \frac{2\alpha^3}{3\sqrt{3}}$$

Per $\alpha = 1$, abbiamo che:

$$\left|\omega\left(\pm\frac{1}{\sqrt{3}}\right)\right| = \frac{2}{3\sqrt{3}} > 0$$

Per $\omega(\pm 1)$ si ha che:

$$\|f - p_2\|_\infty = \frac{1}{6}\frac{2}{3\sqrt{3}} = \frac{1}{9\sqrt{3}} = \frac{\sqrt{3}}{27} \approx 0,067$$

2 Qual è il miglior α?
Sapevamo che $|\omega(\pm\frac{\alpha}{\sqrt{3}})| = \frac{2}{3\sqrt{3}}\alpha^3$, dobbiamo però confrontarlo con il valore che ω assume agli estremi. Cioè:

$$|\omega(\pm 1)| = 1 - \alpha^2$$

Questo valore è ben definito in quanto si ha che è sempre positivo, in quanto $|\alpha| < 1$. Per la definizione di norma infinito segue che:

$$\|\omega\|_\infty = \max\left\{\frac{2}{3\sqrt{3}}\alpha^3, 1 - \alpha^2\right\}$$

Come si comporta però al variare di α?
Sappiamo che la prima quantità corrisponde ad una funzione crescente, mentre la seconda ad una decrescente. Visto che dobbiamo prendere il massimo dei valori delle due funzioni, l'α che ci conviene prendere è quello che fa assumere $\|\omega\|_\infty$ il valore minimo. In questo caso corrisponde con il punto in cui le due funzioni sono uguali. In questo modo quello che stiamo cercando è:

$$\min_\alpha \|\omega(\alpha)\|_\infty$$

Non c'è bisogno di andare a risolvere l'uguaglianza per α, basta far vedere che sostituendo α con il valore proposto, si ottiene che l'uguaglianza è verificata. Facendo i conti, otteniamo $\frac{1}{4}$ da entrambe le parti, quindi p verificata ■

3.7 Forma di Newton

Dalle sezioni precedenti, abbiamo visto che possiamo rappresentare i polinomi con la base delle singole potenze di x $\{x^k : k \in \{0, ..., n\}\}$, oppure in forma di Lagrange $\{L_i(x) : i \in \{0, ..., n\}\}$.

Definizione 3.7.1: Polinomi in forma di Newton

Un polinomio è detto in **Forma di Newton** se è scrivibile come:

$$p_n(x) = a_0 + \sum_{i=1}^{n} a_i \prod_{j=0}^{i-1}(x - x_j)$$

Analizziamo questa definizione. Questo polinomio è equivalente a:

$$p_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n \prod_{k=0}^{n-1}(x - x_k)$$

Con $x_0, ..., x_n$ nodi di interpolazione del polinomio.

Per trovare i coefficienti a_i del polinomio, possiamo trovare l'interpolazione del tipo $p_n(x_i) = y_i$ (se abbiamo una funzione, ci basta porre $y_i = f(x_i)$), ci basta:

$$\begin{aligned} x = x_0 : p_n(x_0) = y_0 \quad &\Rightarrow \quad a_0 = y_0 \\ x = x_1 : p_n(x_1) = y_1 \quad &\Rightarrow \quad a_0 + a_1(x_1 - x_0) = y_1 \\ x = x_2 : p_n(x_2) = y_2 \quad &\Rightarrow \quad a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) = y_2 \end{aligned}$$

E così via per ogni $i \in \{0, ..., n\}$. Notiamo che in questo modo troviamo un sistema lineare triangolare inferiore:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & (x_1 - x_0) & 0 & 0 & \cdots & 0 \\ 1 & (x_2 - x_0) & \prod_{i=0}^{1}(x_2 - x_i) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & & \vdots \\ 1 & (x_n - x_0) & \prod_{i=0}^{1}(x_n - x_i) & \prod_{i=0}^{2}(x - x_i) & \cdots & \prod_{i=0}^{n-1}(x - x_i) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

Osservazione. Il costo computazionale di un sistema lineare triangolare è $(n+1)^2$

Possiamo dare un'altra definizione di polinomio di Newton.

Definizione 3.7.2: Polinomio di Newton

Se definiamo:

$$\varphi_0(x) = 1 \qquad \varphi_i(x) = \prod_{j=0}^{i-1}(x - x_j) \quad i > 0$$

Allora possiamo definire il **Polinomio di Newton** come:

$$p_n(x) = \sum_{i=0}^{n} a_i \varphi_i(x)$$

Abbiamo in questo caso che $\{\varphi_i(x)\}$ è una base di polinomi

Osservazione. Il polinomio $p_{n+1}(x)$ se gli aggiungiamo un nodo x_{n+1} con Lagrange dovremmo ricominciare a calcolarlo da zero. Qui invece possiamo sfruttare il fatto che lo aggiungiamo in fondo:

$$p_{n+1}(x) = a_0 + a_1(x - x_0) + \cdots + a_n \prod_{j=0}^{n-1}(x - x_j) + a_{n+1} \prod_{j=0}^{n}(x - x_j)$$

Cioè, riscrivendola per bene:

$$p_{n+1}(x) = p_n(x) + \omega_n(x)$$

Tornando al sistema lineare, questo implica aggiungere una riga e una colonna in fondo

Osservazione. a_n non dipende dall'ordine dei nodi, in quanto è tutto legato alla produttoria

I coefficienti a_i vengono chiamati anche **Differenza Divisa**, in quanto possono essere ottenuti sfruttando dei "rapporti incrementali". In particolare vale la seguente proposizione:

Proposizione 3.7.3

Posto $f[x_i] = f(x_i)$, si ha che:

$$a_k \equiv f[x_0, ..., x_k] = \frac{f[x_1, ..., x_n] - f[x_0, ..., x_{n-1}]}{x_k - x_0}$$

Di questa proposizione non faremo la dimostrazione. Vediamo però che cosa significa. Sappiamo che per $x = x_0$ si ha che $a_0 \equiv f[x_0]$. Sfruttando la formula, possiamo calcolare a_1 come:

$$a_1 \cong f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0}$$

Sappiamo che $f[x_i] = f(x_i)$. Andando avanti in questo modo, possiamo calcolare $f[x_1, x_2]$ nel seguente modo:

$$f[x_1, x_2] = \frac{f[x_2] - f[x_1]}{x_2 - x_1}$$

Da cui poi a_2 come:

$$a_2 \cong f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$

Proseguendo in questo modo otteniamo tutti i valori a_i

Osservazione. Riprendendo il polinomio in base di Newton, abbiamo che può essere scritto come:

$$\begin{aligned} p_n(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots \\ &= a_0 + (x - x_0)\big(a_1 + (x - x_1)(a_2 + \cdots)\big) \end{aligned}$$

In particolare, per $n = 3$ abbiamo che:

$$p_3(x) = a_0 + (x - x_0)\big(a_1 + (x - x_1)(a_2 + (x - x_2)(a_3))\big)$$

Visto che la valutazione del polinomio può essere estremamente lunga, esiste un metodo più efficace di valutare il polinomio, chiamato **Regola di Horner**.
Chiamiamo come $\pi_n = p_n(\hat{x})$ il valori che p_n assume quando lo valutiamo in $\hat{x}$. Definiamo $\pi_0 = a_n$ e poi continuo andando all'indietro con $\pi_k = (\hat{x} - x_{n-k})\pi_{k-1} + a_{n-k}$ con $k \in \{1, ..., n\}$, cioè (riprendendo il caso $n = 3$)

$$p_3(\hat{x}) = \overbrace{a_0 + (x - x_0)\big(\underbrace{a_1 + (x - x_1)(\overbrace{a_2 + (x - x_2)(\pi_0)}^{\pi_1})}_{\pi_2}\big)}^{\pi_3}$$

Sostanzialmente, questo è quello che c'è dietro alla funzione `polival` di Matlab

4 Interpolazione Composta

Riprendiamo in generale quanto fatto con l'interpolazione. Avevamo che potevamo maggiorare l'errore con:

$$|f(x^*) - p_n(x^*)| \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}(b-a)^{n+1}$$

Esistono altri modi per migliorare ancora questa stima oltre ad aumentare il numero di nodi o cambiandoli direttamente? La cosa migliore sarebbe rimpicciolire l'intervallo, ma come facciamo se vogliamo interpolare esattamente su $[a,b]$? Possiamo suddividere l'intervallo e poi interpolare su ogni sottointervallo ottenuto.
Più formalmente possiamo prendere una suddivisione $\sigma = \{x_0, ..., x_n\} \in \Omega_{a,b}$ tale che:

$$a \equiv x_0 < x_1 < \cdots < x_n \equiv b$$

E poi andare ad interpolare la funzione su ogni intervallo $I_j = [x_j, x_{j+1}]$ per $j \in \{0, ..., n-1\}$. Tutto torna in quanto:

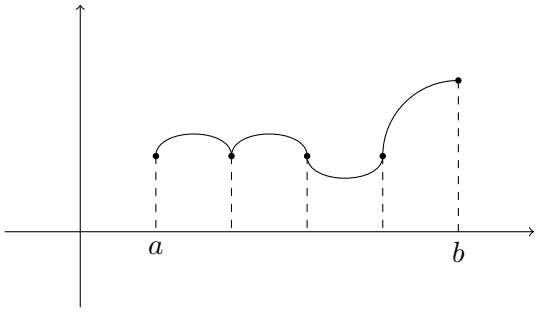
$$[a,b] = \bigcup_{j \in \{0,...,n-1\}} I_j$$

Definiamo poi:

$$\mathcal{X}_{h,\ell} = \{v \in C([a,b]) : v|_{I_j} \in \mathbb{P}_\ell, j \in \{0, ..., n-1\}\}$$

Dove ℓ rappresenta il grado del polinomio, mentre h rappresenta la lunghezza dell'intervallo con lunghezza maggiore.

Osservazione. Possiamo dire che l'interpolazione funziona in quanto v è continua in $[a,b]$, quindi c'è raccordo tra i nodi



Basta che v sia continua e che localmente sia un polinomio

Osservazione. I punti presi in $\sigma \in \Omega_{[a,b]}$ non necessariamente corrispondono con i nodi stessi, alcuni possono essere gli stessi. Volendo per distinguerli potrebbero essere indicati come ξ_i

Possiamo andare a stimare l'errore per ogni intervallino.
Supponiamo $f \in C^{\ell+1}([a,b])$, allora applicando la formula dell'errore sugli intervalli $h_j = |I_j|$ abbiamo che:

$$f(x) - p_{h,\ell}(x) = \frac{f^{(\ell+1)}(\xi_j)}{(\ell+1)!}\omega_\ell(x) \quad x \in I_j$$

Dove abbiamo che p è il polinomio interpolante di f in tale intervallo e ω_ℓ è il polinomio nodale in I_j
Posto:

$$h = \max_j |I_j|$$

Possiamo andare a cercare $P_{h,\ell} \in \mathcal{X}_{h,\ell}$ in modo tale che $P_{h,\ell}|_{I_j}$ corrisponda con il polinomio interpolante. Allora, per $x \in [a,b]$, avremmo che:

$$|f(x) - P_{h,\ell}(x)| \leq \max_{I_j} \frac{|f^{(\ell+1)}(\xi_j)|}{(\ell+1)!} \|\omega_\ell\|_{\infty,I_j}$$

Ma abbiamo che $\|\omega_\ell\|_{\infty,I_j} \leq h^{\ell+1}$, quindi l'errore finale può diventare:

$$|f(x) - P_{h,\ell}(x)| \leq \max_{I_j} \frac{\|f^{(\ell+1)}\|_\infty}{(\ell+1)!} h^{\ell+1}$$

In generale può essere molto molto più potente rispetto al caso generale.

Se poi continuiamo ad aumentare il numero m di sottointervalli, ho che $h \to 0$, quindi l'errore in questo modo va a 0. Così facendo non ho bisogno di aumentare il grado

Osservazione. Aumentando il numero degli intervalli, aumenta anche in maniera considerevole il costo computazionale in termini di valutazione della funzione f, pari a $m \cdot (\ell+1)$ volte

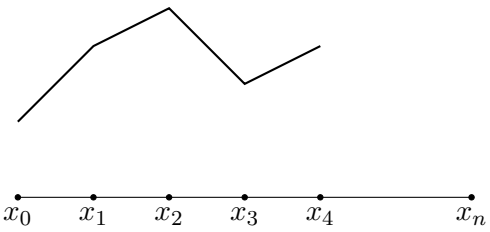
Osservazione. Se i nodi della suddivisione sono equidistanti, allora posso prendere:

$$h = \frac{b-a}{m}$$

Esempio 7 (Caso Lineare)**.** *Il caso lineare è quello che viene fatto da Matlab quindi si usa* `plot`*. In questo caso abbiamo che* $\ell = 1$*, quindi localmente ho:*

$$p_1(x) = y_1 + (x - x_i) + a_i \qquad \text{con } x \in [x_i, x_i + 1], y_i = f(x_i)$$

Graficamente abbiamo che



Localmente utilizzo l'errore di interpolazione e otteniamo che:

$$|f(x) - p_1(x)| = \frac{|f''(\xi_i)|}{2!}|(x - x_i)(x - x_{i+1})| \leq \frac{\|f''\|_{\infty,[x_i,x_i+1]}}{2} \|\omega\|_{\infty,[x_i,x_i+1]}$$

Andiamo a rendere ancora più efficace questa stima raffinando il tutto. Visto che siamo con un polinomio di grado 1*, generalmente possiamo scrivere che* $\|\omega\|_\infty \leq h_i^2$*. Inoltre, visto che il polinomio nodale è una parabola (un polinomio di grado due) che deve passare per* $(x_i, 0)$ *e* $(x_{i+1}, 0)$ *sappiamo che raggiunge il punto di massimo (o di minimo, ma è ininfluente perché prediamo il valore assoluto) precisamente a metà, cioè:*

$$\xi_i = x_{M,i} = \frac{x_i + x_{i+1}}{2}$$

In questo modo, andando a sostituire abbiamo che:

$$|\omega(x_{M,i})| = |(x_{M,i} - x_i)(x_{M,i} - x_{i+1})| = \left| \left(\frac{x_i - x_{i+1}}{2} \right)^2 \right| = \frac{h_1^2}{4} = \|\omega\|_\infty$$

In questo modo abbiamo che:

$$|f(x) - p_1(x)| \leq \frac{\|f''\|_{\infty,[x_i,x_i+1]}}{2}\frac{h_i^2}{4} \qquad per\ x \in [x_i, x_{i+1}]$$

In questo modo abbiamo che:

$$\|f - p_1\|_{\infty,[a,b]} \leq \frac{\|f''\|_{\infty,[a,b]}}{8}h^2$$

Notiamo che è della forma:

$$C \cdot h^{\ell+1}$$

Esercizio. Sia $f(x) = \sin(x)$ con $[a, b] = [-\pi, \pi]$ e con $\ell = 2$.

1. Determina una stime per l'errore
2. Determinare un numero minimo di sottointervalli che assicuri un errore inferiore a 10^{-3}

Soluzione. 1 Per quanto fatto nell'esempio precedente, abbiamo che localmente in $[x_i, x_{i+1}]$:

$$|f(x) - p_2(x)| \leq \frac{\|f'''\|_\infty}{3!}\|\omega\|_\infty$$

Sapendo che $f(x) = \sin(x)$, abbiamo che $\|f'''(x)\| = 1$, quindi:

$$\|f - P_{h,2}\|_{\infty,[pi,\pi]} \leq \frac{1}{6}\|\omega\|_\infty \leq \frac{1}{6}h^3$$

Dove abbiamo usato che $\|\omega\|_\infty \leq h_i^3 \leq h^3$ dove h è il massimo della lunghezza dei sottointervalli.
2 Per avere un errore minore di 10^{-3}, ci basta che sia verificata la condizione:

$$\|f - P_{h,2}\|_\infty \leq \frac{1}{6}h^3 < 10^{-3}$$

Se prendo i nodi equispaziati, ho che:

$$h = \frac{b-a}{m} = \frac{2\pi}{m} \quad \text{Dove } m \text{ è il numero di sottointervalli presi}$$

In questo modo abbiamo che:

$$\frac{1}{6}\left(\frac{2pi}{m}\right)^3 < 10^{-3} \Rightarrow \frac{(2\pi)^3}{m^3} < 6 \cdot 10^{-3} \Rightarrow m^3 > \frac{(2\pi)^3}{6 \cdot 10^{-3}} = \frac{(2\pi)^3 10^3}{6}$$

Cioè abbiamo che:

$$m > \frac{2\pi}{\sqrt[3]{6}}10 \qquad \Rightarrow \qquad m > \left\lceil \frac{2\pi}{\sqrt[3]{6}}10 \right\rceil$$

■

È un'assunzione il fatto che abbiamo preso i nodi equidistanti?

Osservazione. Esiste tutta una classe di metodi che prende il nome di metodi **Adattivi** o **Adattativi** che consiste nel prendere tanti più sottointervalli e con lunghezza minore dove la funzione tende a crescere o decrescere più rapidamente. Nel nostro caso la scelta di nodi equidistanti va più che bene

Come possiamo fare per generalizzare?
Su macchina possiamo effettivamente far vedere che l'ordine di convergenza è pari a $\ell + 1$

Osservazione. In generale, abbiamo che, fissato l'ordine del polinomio composto, si ha che:

$$\|f - P_{h,\ell}\|_\infty \leq C \cdot h^{\ell+1}$$

Quindi si ha convergenza di $P_{h,\ell}$ a f per $h \to 0$ di ordine $\ell + 1$

4.1 Stima dell'Errore di Convergenza

In generale sappiamo che:

$$\|f - P_{h,\ell}\|_\infty \approx C \cdot h^p$$

Vogliamo verificare sperimentalmente che per $h \to 0$ si ha che p tende a $\ell + 1$. Se abbiamo fatto tutto per bene abbiamo che possiamo confrontare $P_{h,\ell}$ con $P_{\frac{h}{2},\ell}$ e possiamo studiarne gli errori. *Devono essere necessariamente h e $\frac{h}{2}$, sennò le cose non tornano.*

$$\frac{E(P_{\frac{h}{2},\ell})}{E(P_{h,\ell})} = \frac{\|f - P_{\frac{h}{2},\ell}\|_\infty}{\|f - P_{h,\ell}\|_\infty}$$

Osserviamo che il polinomio quasi sicuramente non è lo stesso, ma la cosa è del tutto irrilevante. Andando avanti otteniamo che:

$$\frac{\|f - P_{\frac{h}{2},\ell}\|_\infty}{\|f - P_{h,\ell}\|_\infty} = \frac{C(\frac{h}{2})^p}{Ch^p} = \left(\frac{1}{2}\right) p$$

Utilizzando un logaritmo otteniamo che:

$$\log\left(\frac{\|f - P_{\frac{h}{2},\ell}\|_\infty}{\|f - P_{h,\ell}\|_\infty}\right) = p \log\left(\frac{1}{2}\right)$$

Da cui otteniamo che:

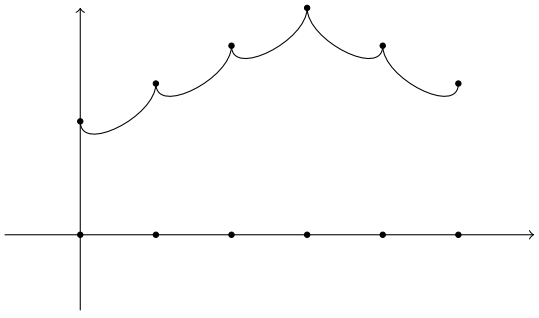
$$p = \frac{1}{\log(\frac{1}{2})} \log\left(\frac{\|f - P_{\frac{h}{2},\ell}\|_\infty}{\|f - P_{h,\ell}\|_\infty}\right) \xrightarrow{h\to 0} \ell + 1$$

Se così non dovesse essere, allora ci sono due possibilità:

- Abbiamo sbagliato a fare i conti da qualche parte
- Potrebbe non valere la relazione, per esempio se f non è sufficientemente regolare

4.2 Splines

Nel corso degli ultimi 50 anni si è voluto trovare una funzione che non solo fosse continua, ma anche regolare. *In sintesi si voleva evitare:*



Questa è continua ma evidentemente è non regolare
Si è voluto quindi creare le **Splines** per ovviare questo problema.

Definizione 4.2.1: Splines

Dati $x_0, x_1, ..., x_k$ $k+1$ nodi distinti ordinati in $[a,b]$, si dice **Spline** di grado ℓ relativa ai nodi x_i una funzione s_ℓ tale che:

$$s_\ell|_{[x_i,x_{i+1}]} \in \mathbb{P}_l \qquad s_\ell \in C^{\ell-1}([a,b])$$

Possiamo indicare con $S_\ell = \{s_\ell\}$ lo spazio delle splines di grado ℓ

Andiamo a studiare i gradi di libertà dello spazio.
Abbiamo $\ell+1$ coefficienti su k intervalli (ogni intervallo ha $\ell+1$ coefficienti e ho k intervalli) quindi iniziamo con l'avere $(\ell+1)\cdot k$ variabili. Sui $k-1$ nodi interno ho delle ℓ condizioni, cioè il fatto che s_ℓ deve essere $C^0, C^1, ..., C^{\ell-1}$, cioè, le derivate i-esime destre e sinistre di ogni punto devono coincidere:

$$p_j^{(m)}(x_i) = p_{j+1}^{(m)}(x_i)$$

Dove $j \in \{0, ..., k-1\}$ indica il polinomio associato all'intervallo j-esimo, $m \in \{0, ..., \ell-1\}$ indica l'ordine della derivata e $i \in \{1, ..., k-1\}$ indica il punto nodale

Mettendo tutto insieme abbiamo quindi che ogni funzione in S_ℓ ha $k+\ell$ coefficienti ancora liberi. Nell'esempio 7, che può essere visto come spline con $\ell = 1$, abbiamo una soltanto condizione libera, che corrisponde a a_1.

Per il nostro problema, vogliamo scegliere una $s_\ell \in S_\ell$ che interpoli i dati, cioè:

$$s_\ell(x_i) = f(x_i) \qquad \forall i \in \{0, ..., k\}$$

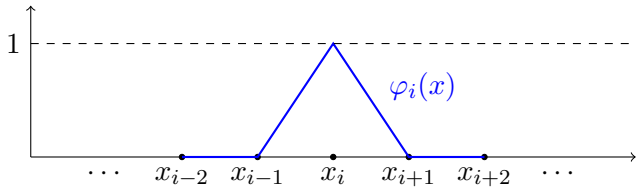
Imporre tutto questo significa imporre altre $k+1$ condizioni, per cui arriviamo ad un totale di $\ell-1$ gradi di libertà, quindi ci sono un'infinità di funzioni che possono andare bene.

Come possiamo trovare quella che vada per noi? Cioè come possiamo saturare le condizioni in modo da trovare quella giusta per noi? Ci sono varie possibilità:

- **Spline Periodica**: Poniamo la condizione $s_\ell^{(j)}(a) = s_\ell^{(j)}(b), j \in \{1, ..., \ell-1\}$
- **Spline Naturali**: Poniamo la condizione $s_\ell^{(j)}(a) = 0 = s^{(j)}(b)$ per certi j in modo che da avere $\ell-1$ condizioni. A volte può capitare che questo tipo di spline possa portare a cose incoerenti, per esempio gli estremi schiacciati quando non dovrebbe essere.

4.3 Splines Lineare

Il caso delle splines lineare è il caso in cui si ha $\ell = 1$. In questo caso la spline è univocamente determinata dal fatto che deve interpolare f nei $k+1$ nodi. Questo in un certo senso lo avevamo già fatto, in quanto questo corrisponde ad un polinomio lineare composto, che è esattamente quanto già visto in precedenza.
Graficamente, quello che le spline $\ell = 1$ è:



La funzione $\varphi_i(x)$ prende il nome di "*Hat function*" perché appunto sembra un cappellino. Notiamo poi che questo tipo di funzione è caratterizzata di essere continua e non nulla in un determinato intervallo. In particolare:

$$\varphi_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{x_i - x_{i-1}} & \text{per } x \in [x_{i-1}, x_i] \\ \dfrac{x_{i+1} - x}{x_{x+1} - x_i} & \text{per } x \in [x_i, x_{i+1}] \\ 0 & \text{altrimenti} \end{cases}$$

Queste φ_i hanno una regione piccola in cui non sono nulle.
Inoltre posso definirle anche per φ_k che ha solo la parte a sinistra (quella crescente) e φ_0 che ha solo quella destra (decrescente). Ma in questo modo posso definirle per tutte.

Volendo si può dimostrare che queste funzioni rappresentano una base su $[a, b]$ per S_1, per cui abbiamo che:

$$s_1 = \sum_{i=0}^{k} \alpha_i \varphi_i(x)$$

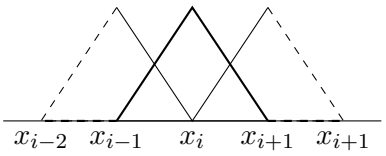
Prendendo la condizione di interpolazione, abbiamo che e notando che $\varphi_i(x_j) = \delta_{i,j}$ (*Secondo la delta di Kronecker*), abbiamo che:

$$f(x_j) = s_1(x_j) = \sum_{i=0}^{k} \alpha_i \varphi_i(x_j) = \alpha_j$$

In maniera del tutto naturale segue che:

$$s_1(x) = \sum_{i=0}^{k} f(x_i)\varphi(x)$$

Rispetto a quanto veniva fatto con i polinomi di Lagrange e altro ancora, abbiamo che questi polinomi sono delle funzioni locali. Questa cosa, quando si hanno delle perturbazioni, ha delle conseguenze fenomenali, perché permette una minore proliferazione delle perturbazioni sull'intero intervallo.



Possiamo notare con estrema semplicità che le funzioni coinvolte sono solamente 3. Non c'è propagazione di errore

4.4 Splines Cubiche

Le spline cubiche (cioè con $\ell = 3$) sono C^2, quindi sono sufficientemente lisce e con calcolo non troppo costoso.

Definiamo $M_i = s_3''(x_i) = s''(x_i)$ per $i \in \{0, ..., k\}$. Queste saranno le nostre incognite. Notiamo che se la spline s è un polinomio di terzo grado, allora la su derivata seconda è un polinomio di primo grado. Possiamo quindi definire:

$$s''(x) = M_i \frac{x - x_{i-1}}{x_i - x_{i-1}} + M_{i-1} \frac{x_i - x}{x_i - x_{i-1}} \quad \text{per } x \in [x_{i-1}, x_i]$$

Per questioni di comodità, possiamo porre $h_i = x_i - x_{i-1}$. Integrando su quanto appena trovato, troviamo che:

$$s'(x) = \frac{M_i}{h_i}\frac{1}{2}(x - x_{i-1})^2 - \frac{M_{i-1}}{h_i}\frac{1}{2}(x_i - x)^2 c_i$$

Integrando nuovamente abbiamo che:

$$s(t) = \frac{M_i}{h_i}\frac{1}{6}(x - x_{i-1})^3 + \frac{M_{i-1}}{h_i}\frac{1}{6}(x_i - x)^3 + c_i(x - x_{i-1}) + \tilde{c}_i$$

Cerchiamo di eliminare c e $\tilde{c}_i$ imponendo la condizione di interpolazione. Cioè imponiamo che:

$$s(x_{i-1}) = f(x_{i-1}) = f_{i-1}$$

Allora in questo modo abbiamo che:

$$\underbrace{\frac{M_i}{h_i}\frac{1}{6}(x_{i-1} - x_{i-1})^3}_{0} + \frac{M_{i-1}}{h_i}\frac{1}{6}(\underbrace{x_i - x_{i-1}}_{h_i})^3 + c_i(\underbrace{x_{i-1} - x_{i-1}}_{0}) + \tilde{c}_i = f_{i-1}$$

Da cui segue che:

$$\tilde{c}_i = f_{i-1} - \frac{1}{6}h_i^2 M_{i-1}$$

Andiamo ad imporre l'altra condizione, cioè $s(x_i) = f_i$. In questo modo otteniamo che:

$$\frac{M_i}{h_i}\frac{1}{6}h_i^3 + 0 + c_i h_i + \tilde{c}_i = f_i$$

Da cui si ottiene che:

$$h_i c_i = f_i - \frac{1}{6}h_i^2 M_i - \tilde{c}_i = f_i - f_{i-1} + \frac{1}{6}h_i^2 M_{i-1} - \frac{1}{6}h_1^2 M_i$$

Da cui:

$$c_i = \frac{f_i - f_{i-1}}{h_i} = \frac{1}{6}h_i M_i - \frac{1}{6}h_i M_i$$

Andiamo ad imporre adesso la condizione di regolarità in modo da determinare i vari M_i, cioè imponiamo:

$$s'\big|_{[x_{i-1},x_i]}(x_i) = s'\big|_{[x_i,x_{i+1}]}(x_i) \qquad \forall i \in \{1, ..., k-1\}$$

Andando a sostituire l'espressione in s' ad inizio pagina otteniamo che:

$$s'(x) = \frac{M_i}{h_i}\frac{1}{2}(x - x_{i-1})^2 - \frac{M_{i-1}}{h_i}\frac{1}{2}(x_i - x)^2 + c_i|_{[x_{i-1},x_i]}$$
$$s'(x) = \frac{M_{i+1}}{h_{i+1}}\frac{1}{2}(x - x_i)^2 - \frac{M_i}{h_{i+1}}\frac{1}{2}(x_{i+1} - x)^2 + c_{i+1}|_{[x_i,x_{i+1}]}$$

Andando ad imporre l'uguaglianza in x_i abbiamo che:

$$\frac{M_i}{h_i}\frac{1}{2}h_i^2 + \frac{f_i - f_{i-1}}{h_i} + \frac{1}{6}h_i M_{i-1} - \frac{1}{6}h_i M_i = -\frac{1}{2}\frac{M_i}{h_{i+1}}h_{i+1}^2 + \frac{f_{i+1} - f_i}{h_{i+1}} + \frac{1}{6}h_{i+1}M_i - \frac{1}{6}h_{i+1}M_{i+1}$$
$$\frac{1}{3}h_i M_i - \frac{1}{6}h_i M_{i-1} + \frac{f_i - f_{i-1}}{h_i} = -\frac{1}{3}h_{i+1}M_i - \frac{1}{6}h_{i+1}M_{i+1} + \frac{f_{i+1} - f_i}{h_{i+1}}$$

In questo modo otteniamo che:

$$\frac{1}{6}h_I M_{i-1} + \left(\frac{1}{3}h_i + \frac{1}{3}h_{i+1}\right) M_i + \frac{1}{6}h_{i+1}M_{i+1} = \frac{f_{i+1} - f_i}{h_{i+1}} - \frac{f_i - f_{i-1}}{h_i}$$

Poiché questo è valido per ogni $i \in \{1, ..., k-1\}$, abbiamo sostanzialmente un sistema lineare della forma:

$$\begin{pmatrix} \ddots & \ddots & \ddots & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \frac{1}{6}h_i & \frac{1}{3}(h_i + h_{i+1}) & \frac{1}{6}h_i & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \ddots & \ddots & \ddots \end{pmatrix} \begin{pmatrix} M_0 \\ \vdots \\ M_{i-1} \\ M_i \\ M_{i+1} \\ \vdots \\ M_k \end{pmatrix} = \begin{pmatrix} \\ \vdots \\ \\ \frac{f_{i+1}-f_i}{h_{i+1}} - \frac{f_i - f_{i-1}}{h_i} \\ \\ \vdots \\ \\ \end{pmatrix}$$

Questo è un sistema lineare $(k-1) \times (k+1)$, cioè abbiamo $k-1$ condizioni per $k+1$ incognite, infatti abbiamo $\ell - 1 = 3 - 1 = 2$ gradi di libertà ancora da decidere per trovare s_3 in maniera univoca.

4.5 Completamento delle Condizioni

Abbiamo diversi modi per riempire le due condizioni:

- **Spline Naturale**: In questo modo imponiamo $s_3''(a) = 0 = s_3''(b)$, in questo modo la funzione tende ad appiattirsi agli estremi. Ovviamente ha senso fare questa cosa se la funzione f ha già questa caratteristica. Matematicamente parlando, questo è equivalente a imporre:

  $$s_3''(a) = M_0 = 0 \qquad \text{e} \qquad s_3''(b) = M_k = 0$$

  In questo modo abbiamo che tutte le condizioni che ci servono per un sistema lineare non sovradeterminato (bisognerebbe dimostrare che non è singolare, ma va bene così)

- **Spline Completa** o **Vincolata**: Supponiamo di conoscere $f'(a)$ e $f'(b)$ e ritorniamo nuovamente a s' con $i = 1$:

  $$s'(x) = \frac{M_i}{h_i}\frac{1}{2}(x - x_{i-1})^2 - \frac{M_{i-1}}{h_i}\frac{1}{2}(x_i - x)^2$$

  Supponiamo $s'(x_0) = f'(x_0)$, allora abbiamo che:

  $$\begin{aligned} \frac{M_1}{h_1}\frac{1}{2}0 - \frac{M_0}{h_1}\frac{1}{2}h_1^2 + \frac{f_1 - f_0}{h_1} + \frac{1}{6}h_1 M_0 - \frac{1}{6}h_1 M_1 &= f'(x_0) \\ \left(\frac{1}{6}h_1 - \frac{1}{2}h_1\right) M_0 - \frac{1}{2}h_1 M_1 &= f'(x_0) - \frac{f_1 - f_0}{h_1} \end{aligned}$$

  In questo modo abbiamo aggiunto una riga in cima alla matrice. Facendo poi una cosa simile per $i = k$, possiamo aggiungere una riga anche in fondo, in modo da avere un sistema lineare con matrice quadrata tridiagonale.

- **"Not-a-Knot"**: In questo modo aggiungo un'ulteriore condizione su s_3 negli intervalli $[x_0, x_1]$ e $[x_{k-1}, x_k]$. La condizione aggiuntiva che andiamo ad imporre è:

$$s'''\big|_{[x_0,x_1]}(x_1) = s'''\big|_{[x_1,2]}(x_1)$$

Questa è una condizione che va a lavorare direttamente su M_0 e su M_1. In maniera del tutto analoga possiamo imporre:

$$s'''\big|_{[x_{k-2},x_{k-1}]}(x_{l-1}) = s'''\big|_{[x_{k-1},x_k]}(x_{k-1})$$

Poi sarebbe da controllare che i sistemi lineari ottenuti non siano singolari. *Questo è quello che fa Matlab*

4.6 Risultati di Convergenza e Regolarità

Teorema 4.6.1

Sia $f : [a, b] \to \mathbb{R}$ e s_3 spline cubica naturale ($s_3''(a) = 0 = s_3''(b)$) o completa ($s'(a) = f'(a)$ e $s'(b) = f'(b)$) interpolante nei nodi assegnati in $[a, b]$. Allora $\forall g \in C^2([a, b])$ interpolante f negli stessi nodi, che sia naturale o completa, vale:

$$\int_a^b \big(s_3''(x)\big)^2\,dx \le \int_a^b \big(g''(x)\big)\,dx$$

L'uguaglianza nel caso di spline completa si ha se e solo se $g = s_3$

Considerazioni Aggiuntive. *Quello che sostanzialmente il teorema ci dice è che qualunque funzioni $g \in C^2$ che abbia le stesse proprietà di s_3, s_3 è la più liscia di tutte. Lo si può vedere in particolare dal valore dell'integrale in quanto, rappresentando l'area sottesa dal grafico, più è piccola, più è schiacciata. In un certo senso, il valore dell'integrale al quadrato può essere interpretato come una norma tra funzioni, di cui s_3 ha la norma minore*

Dimostrazione. Come prima cosa facciamo vedere l'uguaglianza:

$$(\star) \qquad \int_a^b (g''(x))^2 dx = \int_a^b (g''(x) - s''(x))^2 dx + \int_a^b (s''(x))^2 dx$$

In tal caso abbiamo che vale l'uguaglianza presente nel teorema, in quanto abbiamo sicuramente che:

$$\int_a^b (g''(x) - s''(x))^2 dx \ge 0$$

Inoltre, assumendo che valga $(\star)$, vediamo il caso dell'uguaglianza. Abbiamo che:

$$\int_a^b s''(x)dx = \int_a^b g''(x)dx \qquad \Leftrightarrow s''(x) = g''(x)$$

Ma da questo abbiamo anche che questi due valori sono uguale se e solo se:

$$\forall x \in [a, b] \quad s'(x) = g'(x) + c$$

Assumendo una spline completa, abbiamo che:

$$s'(a) = f'(a) = g'(a) + c \qquad \Rightarrow \qquad c = 0$$

Da cui segue che $s'(x) = g'(x), \forall x \in [a,b]$. Continuando ad integrare abbiamo che:

$$s'(x) = g'(x) \qquad \Rightarrow \quad s(x) = g(x) + c_1$$

Ma abbiamo che sia s sia g sono dei polinomi che interpolano f nei nodi x_i, quindi:

$$\forall x_i \quad s(x_i) = f(x_i) = g(x_i) + c_1 \qquad \Rightarrow \qquad c_1 = 0$$

Da cui segue l'uguaglianza dei due polinomi $s(x) = g(x), \forall x \in [a,b]$

Per verificare effettivamente la veridicità della disuguaglianza del teorema, mostriamo che l'uguaglianza ($\star$) è effettivamente vera, cioè, dimostriamo:

$$\int_a^b (g''(x))^2 dx = \int_a^b (g''(x) - s''(x))^2 dx + \int_a^b (s''(x))^2 dx$$

Quindi:

$$\begin{aligned}
\int_a^b (g''(x))^2 dx &\stackrel{?}{=} \int_a^b (g''(x) - s''(x))^2 dx + \int_a^b (s''(x))^2 dx \\
\int_a^b (g''(x))^2 dx &\stackrel{?}{=} \int_a^b (g''(x))^2 + (s''(x))^2 - 2g''(x)s''(x) dx + \int_a^b (s''(x))^2 dx \\
0 &\stackrel{?}{=} \int_a^b 2(s''(x))^2 - 2g''(x)s''(x) dx \\
0 &\stackrel{?}{=} \int_a^b (s''(x))^2 - 2g''(x)s''(x) dx \\
0 &\stackrel{?}{=} \int_a^b s''(x)(s''(x) - g''(x)) dx \\
0 &\stackrel{?}{=} \left[s''(x)(s'(x) - g'(x))\right]_a^b - \int_a^b s'''(x)(s'(x) - g'(x)) dx
\end{aligned}$$

Mostriamo ora che entrambi i membri sono nulli. Il primo membro è uguale a:

$$s''(b)(s'(b) - g'(b)) - s''(a)(s'(a) - g'(a))$$

Se siamo nel caso di una spline naturale, allora abbiamo che $s''(a) = s''(b) = 0$, per cui è uguale a 0. Se invece siamo nel caso di una spline completa, allora abbiamo che $s'(b) = f'(b) = g'(b)$ e $s'(a) = f'(a) = g'(a)$, per cui anche il secondo membro è uguale a 0.
Ci resta da dimostrare che anche l'altra parte è nulla:

$$\int_a^b s'''(x)(s'(x) - g'(x)) dx = \sum_{i=0}^{k-1} \int_{x_i}^{x_{i+1}} s'''(x)(s'(x) - g'(x)) dx$$

Ricordando tuttavia che la spline era di terzo grado, abbiamo che $\forall i, \forall [x_i, x_{i+1}]$ si ha che s''' è costante, quindi possiamo portarlo fuori dall'integrale:

$$\begin{aligned}
\int_a^b s'''(x)(s'(x) - g'(x)) dx &= \sum_{i=0}^{k-1} \int_{x_i}^{x_{i+1}} s'''(x)(s'(x) - g'(x)) dx \\
&= \sum_{i=0}^{k-1} s'''(x_i) \int_{x_i}^{x_{i+1}} (s'(x) - g'(x)) dx \\
&= \sum_{i=0}^{k-1} s'''(x_i)[s(x) - g(x)]_{x_i}^{x_{i+1}}
\end{aligned}$$

Abbiamo però che $\forall i, s(x_i) = f(x_i) = g(x_i)$ e $s(x_{i+1}) = f(x_{i+1}) = g(x_{i+1})$, quindi ogni termine della somma è nullo e di conseguenza anche la somma. Quindi abbiamo verificato $(\star)$ e di conseguenza anche la disuguaglianza del teorema. □

4.7 Risultati di Convergenza

Teorema 4.7.1

Sia $f \in C^4([a,b])$ e si consideri una partizione $\sigma \in \Omega_{a,b}$ di ampiezza h_i. Sia s_3 la spline cubica interpolante f completa (cioè che $s_3'(a) = f'(a)$ e $s_3'(b) = f'(b)$). Allora abbiamo che:

$$\|f^{(r)} - s_3^{(r)}\|_\infty \leq c_r h^{4-r} \|f^{(4)}\|_\infty \qquad r \in \{0,1,2,3\}$$

Con $c_0 = \frac{5}{384}$, $c_1 = \frac{1}{24}$, $c_2 = \frac{3}{8}$ e $c_4 = \frac{1}{2}(\beta - \frac{1}{\beta})$ dove $\beta = \frac{4}{\min h_i}$ e $h = \max h_i$

Considerazioni Aggiuntive. *c_3 è l'unica costante che non varia in base alla scelta degli intervalli*

Dimostrazione. Per $r = 0$ avevamo da teoremi precedenti che:

$$\|f - s_3\|_\infty \leq c_0 h^4 \|f^{(4)}\|_\infty$$

Questo è un risultato che ci aspettavamo con i polinomi di grado 3, quindi s_3 approssima come h^4 Ma le spline non solo approssimano bene la funzione, ma le sue derivare approssimano bene le derivate della funzione stessa. In particolare abbiamo che:

$$\|f' - s_3'\|_\infty \leq c_1 h^3 \|f^{(4)}\|_\infty \qquad r = 1$$

Quindi anche la pendenza della funzione è ben approssimata. Ma non solo, anche la sua curvatura è ben approssimata, in quanto:

$$r = 2 \qquad \|f'' - s_3'' \leq c_2 h^2 \|f^{(4)}\|_\infty$$

Normalmente con gli altri polinomi ci si fermerebbe solo al primo. □

5 Approssimazione di Integrali

5.1 Presentazione del Problema

Quello di cui ci occuperemo in questo capitolo è: Data una funzione $f : [a, b] \to \mathbb{R}$ funzione integrabile, io voglio approssimare

$$\mathcal{I}(f) = \int_a^b f(x)dx$$

Sarà possibile fare questo attraverso Formule di Quadratura e Interpolazione Numerica.

Vogliamo quindi determinare $f_n \approx f$ dove n è un qualche parametro e definiamo quindi:

$$\mathcal{I}_n(f) = \mathcal{I}(f_n) = \int_a^b f_n(x)dx$$

In generale le formule di quadratura porteranno a formule del tipo:

$$\mathcal{I}(f_n) = \sum_{i=0}^{n} \alpha_n f(x_i)$$

Notiamo che questa è molto simile ad una media pesata, dove gli α_i sono i pesi e $f(x_i)$ sono gli elementi di cui vogliamo trovare la media. Notiamo anche che assomiglia molto anche ad un'interpolazione, infatti possiamo definire x_i come i nodi. *Infatti questa non è altro che un'applicazione dell'interpolazione*

In generale avremo che l'errore sarà definito come:

$$E_n(f) = \mathcal{I}(f) - \mathcal{I}(f_n) = \int_a^b f(x) - f_n(x)dx$$

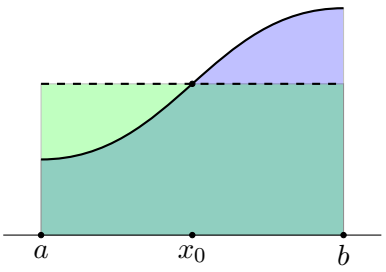
In particolare, l'errore è un numero reale, quindi facendone il valore assoluto abbiamo che:

$$|E_n(f)| \leq \int_a^b |f(x) - f_n(x)|dx \leq (b-a)\|f - f_n\|_\infty$$

Vediamo adesso delle formule di quadratura

5.2 Formula del Rettangolo o del Punto Medio

Prendiamo il polinomio interpolante di grado $n = 0$ tale che $p_o(x) = f(x_0)$ dove x_0 è il punto medio dell'intervallo $[a, b]$. Graficamente avremmo allora che:



Abbiamo quindi che:

$$\mathcal{I}_0(f) = \int_a^b f(x_0)dx = f(x_0)(b-a)$$

Proposizione 5.2.1

Sia $f \in C^2([a,b])$, allora:

$$E_0 = \frac{1}{3}\left(\frac{b-a}{2}\right)^3 f''(\xi) \qquad \xi \in (a,b)$$

Considerazioni Aggiuntive. *Notiamo che l'errore dipende dalla regolarità di f. Infatti questo risultato è vero se $f \in C^2$ almeno. Altrimenti la cosa non è vera. Nulla vieta però che possa essere comunque utilizzato come metodo di approssimazione.*

Dimostrazione. Sviluppiamo f in un intorno di x_0 con Taylor. Otteniamo allora che:

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + \frac{1}{2}f''(\xi_x)(x-x_0)^2$$

Integrando abbiamo che:

$$E_0(f) = \int_a^b f(x) - f(x_0)dx = \int_a^b f'(x_0)(x-x_0)dx + \int_a^b \frac{1}{2}f''(\xi_x)(x-x_0)^2dx$$

Andiamo a risolvere i due integrali, uno per volta:

$$\begin{aligned}\int_a^b f'(x_0)(x-x_0)dx &= \left[f'(x_0)\frac{1}{2}(x-x_0)^2\right]_a^b = \frac{f'(x_0)}{2}\left[\left(b-\frac{a+b}{2}\right)^2 - \left(a-\frac{a+b}{2}\right)^2\right] \\ &= \frac{(f'(x_0))}{2}\left[\left(\frac{b-a}{2}\right)^2 - \left(\frac{b-a}{2}\right)^2\right] = 0\end{aligned}$$

Per poter risolvere l'altro integrale con facilità, ci tornerà questo lemma:

Lemma 5.2.2: Media Integrale

Sia f continua in $[a,b]$ e sia g integrabile in $[a,b]$ con stesso segno in $[a,b]$ (cioè $g \leq 0$ oppure $g \geq 0$ in tutto l'intervallo). Allora:

$$\exists \xi \in]a,b[: \int_a^b f(x)g(x)dx = f(\xi)\int_a^b g(x)dx$$

Dimostrazione. Senza perdere di generalità supponiamo $g(x) \geq 0, \forall x \in [a,b]$. Allora abbiamo che:

$$\begin{aligned}\int_a^b f(x)g(x)dx &\leq \int_a^b \left(\max_{x\in[a,b]} f(x)\right) g(x)dx = \max_{x\in[a,b]} f(x)\int_a^b g(x)dx \\ \int_a^b f(x)g(x)dx &\geq \int_a^b \left(\min_{x\in[a,b]} f(x)\right) g(x)dx = \min_{x\in[a,b]} f(x)\int_a^b g(x)dx\end{aligned}$$

In questo modo abbiamo che:

$$\min_{x\in[a,b]} f(x) \leq \frac{\displaystyle\int_a^b f(x)g(x)dx}{\displaystyle\int_a^b g(x)dx} \leq \max_{x\in[a,b]} f(x)$$

Possiamo dire questa cosa in quanto abbiamo che g per ipotesi ha sempre lo stesso segno, quindi ha integrale non nullo. Se fosse stato nullo, allora anche g era nulla, ma in tal caso il lemma sarebbe stato banalmente verificato. Sapendo che f è una funzione continua, $\exists \xi \in]a,b[$ tale che:

$$\frac{\int_a^b f(x)g(x)dx}{\int_a^b g(x)dx} = f(\xi)$$

A questo punto si moltiplica e abbiamo finito □

Eravamo rimasti che volevamo risolvere l'integrale:

$$\int_a^b \frac{1}{2} f''(\xi_x)(x-x_0)^2 dx$$

Applichiamo il lemma, allora abbiamo che:

$$\int_a^b \frac{1}{2} f''(\xi_x)(x-x_0)^2 dx = \frac{1}{2} f''(\xi) \int_a^b (x-x_0)^2 dx = \frac{1}{2} f''(\xi) \left[\frac{1}{3}(x-x_0)^3\right]_a^b$$

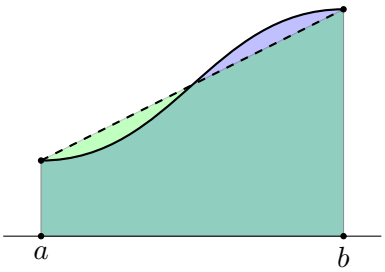
A questo punto basta fare i conti e ritorna quanto annunciato nel teorema.
Notiamo che $f''(\xi_x)$ è continua in quanto per lo sviluppo di Taylor fatto a inizio dimostrazione abbiamo che $f''(\xi_x)$ è somma di funzioni continue. □

5.3 Formula del Trapezio

In questo caso abbiamo $n = 1$, quindi abbiamo un polinomio lineare con nodi $x_0 = a$ e $x_1 = b$. Prendiamolo come polinomio di Lagrange, quindi:

$$p_1(f) = L_0(x)f(x_0) + L_1(x)f(x_1) = f(a)\frac{x-b}{a-b} + f(b)\frac{x-a}{(}b-a)$$

Graficamente abbiamo che:



Andando quindi a calcolare l'integrale abbiamo che:

$$\begin{aligned}
\mathcal{I}_1(f) &= \int_a^b p_1(x)dx = \frac{f(a)}{a-b}\int_a^b (x-b)dx + \frac{f(b)}{b-a}\int_a^b (x-a)dx \\
&= \frac{f(a)}{a-b}\left[\frac{1}{2}(x-b)^2\right]_a^b + \frac{f(b)}{b-a}\left[\frac{1}{2}(x-a)^2\right]_a^b = \frac{1}{2}\frac{f(a)}{a-b}(-(a-b))^2 + \frac{1}{2}\frac{f(b)}{b-a}(b-a)^2 \\
&= \frac{1}{2}\frac{(b-a)^2}{b-a}(f(a)+f(b)) = \frac{(b-a)(f(a)+f(b))}{2}
\end{aligned}$$

Proposizione 5.3.1

Se $f \in C^2([a,b])$, allora:

$$E_1(f) = -\frac{(b-a)^3}{12} f''(\xi) \qquad \xi \in (a,b)$$

Considerazioni Aggiuntive. *Aumentando di grado non abbiamo guadagnato nulla*

Dimostrazione. Per p_1 polinomio di Lagrange vale:

$$f(x) - p(x) = \frac{f''(\xi_x)}{2}\omega(x)$$

Dove ω è il polinomio nodale $\omega = (x-a)(x-b)$. Ma $\omega(x) \leq 0$ per $x \in [a,b]$. Sapendo poi che $E_n(f) = \mathcal{I}(f) - \mathcal{I}(p_n)$ abbiamo che:

$$E_1(f) = \int_a^b (f(x) - p_1(x))dx = \int_a^b \frac{f''(\xi_x)}{2}\omega(x)dx$$

Utilizzando il lemma 5.2.2, abbiamo che $\exists \xi \in [a,b]$ tale che:

$$E_1 = \frac{f''(\xi)}{2}\int_a^b (x-a)(x-b)dx = [...] = -\frac{(b-a)^3}{12}f''(\xi)$$

□

5.4 Formula di Cavalieri - Simpson

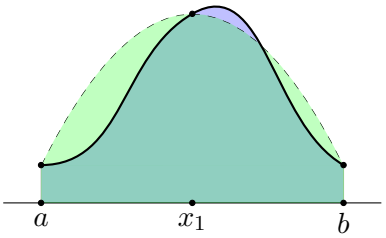
Qui prendiamo tre nodi, il punto a, il punto b e il punto medio. Usiamo p_2 polinomio di Lagrange di grado 2. Seguendo gli ragionamenti visti abbiamo che:

$$\mathcal{I}_2 = \frac{b-a}{2}f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)$$

Otteniamo quindi che l'errore è:

$$E_2 = -\frac{1}{90}\left(\frac{b-a}{2}\right)^5 f^{(4)}(\xi)$$

Questo nell'eventualità che $f \in C^4([a,b])$



Possiamo notare subito che c'è un enorme salto di qualità, abbiamo infatti un grado 5 per l'intervallo e un grado 4 per la derivata della funzione.

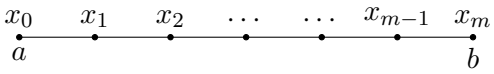
Definizione 5.4.1: Grado di Precisione

Una formula di quadratura ha **Grado di Precisione** k se è esatta per polinomi di grado al più k

Confrontando quanto fatto con le altre formule abbiamo immediatamente che sia la Formula del Rettangolo (p_0 di Lagrange) sia la Formula del Trapezio (p_1 di Lagrange) hanno grado di precisione 1 (con l'errore misurato con la derivata seconda). Mentre la formula di Cavalieri-Simpson (p_2 di Lagrange) ha grado di precisione 3 (errore regolato dalla derivata quarta).
Volendo può esser dimostrato che c'è una correlazione con il grado k del polinomio: se il grado è k pari, allora si ha un grado di precisione $k+1$, altrimenti si ha un grado k. *Ne consegue che è più conveniente lavorare con polinomi di grado pari.*

5.5 Formule Quadratiche Composte

Tutti gli errori delle formule viste fin'ora hanno una parte del tipo $(b-a)^k$ per opportuno k. Possiamo allora immaginare di suddividere l'intervallo in tanti sottointervalli e di proseguire poi come con l'interpolazione.



Prendiamo tutti nodi equidistanti, quindi avremo che la loro distanza, o equivalentemente la lunghezza dei vari intervallini sarà:

$$H = |I_i| = fracb - am$$

Vediamo ora caso per caso come adattare le formule precedenti al caso composto.

Formula dei Rettangoli Composta

Se $\{x_i\}_{i\in\{0,\dots,m\}}$ sono i nodi, poniamo come $\{\hat{x}_i\}_{i\in\{1,\dots,m\}}$ i punti medi dei rispettivi intervalli i-esimi. Allora abbiamo che:

$$\mathcal{I}_{0,m} = Hf(\hat{x}_1) + Hf(\hat{x}_2) + \cdots + Hf(\hat{x}_n) = H(f(\hat{x}_1) + \cdots + f(\hat{x}_n))$$

In Matlab può essere scritto facilmente come `H*sum(|f(ξ)|)`. Vediamo adesso l'errore:

$$E_{0,m}(f) = \sum_{i=1}^{m} E_0(f)|_{I_i} = \sum_{i=1}^{m} \frac{1}{24}H^3 f''(\xi_i) = \frac{1}{24}H^3 \sum_{i=1}^{m} f''(\xi_i)$$

Per $\xi_i \in I_i$. Ma sappiamo anche che $H = \frac{b-a}{m}$, per cui:

$$E_{0,m}(f) = \frac{1}{24}H^3 \sum_{i=1}^{m} f''(\xi_i) = \frac{H^2}{24}(b-a)\cdot\frac{1}{m}\sum_{i=1}^{m} f''(\xi_i)$$

Ma questa adesso è una media, allora, se $f \in C^2([a,b])$ possiamo utilizzare la versione discreta della media integrale 5.2.2 e abbiamo che $\exists \xi \in]a,b[$ tale che:

$$\frac{1}{m}\sum_{i=1}^{m} f''(\xi_i) = f''(\xi)$$

Per cui, andando a sostituire:

$$E_{0,m} = \frac{H^2}{24}(b-a)f''(\xi)$$

Formula dei Trapezi Composta
Qui abbiamo che i nodi sono gli stessi, non c'è bisogno di aggiungerne altri. L'integrale diventa quindi:

$$\mathcal{I}_{1,m} = \sum_{i=0}^{m-1} \frac{H}{2}(f(x_i)+f(x_{i+1})) = \frac{1}{2}H(\underbrace{f(x_0)+f(x_1)}_{I_1} + \underbrace{f(x_1)+f(x_2)}_{I_2} + \cdots + f(x_{m-1}) + f(x_m))$$

I nodi centrali sono presi due volte, quindi possiamo scrivere:

$$\mathcal{I}_{1,m} = \frac{1}{2}H(f(x_0) + 2f(x_1) + \cdots + 2f(x_{m-1}) + f(x_m))$$

L'errore, seguendo gli stessi passaggi fatti prima, è:

$$E_{1,m} = -\frac{b-a}{12}H^2 f''(\xi) \qquad \xi \in]a,b[$$

Formula di Cavalieri-Simpson Composta
Anche qui, come nel caso della formula normale, per ogni sottointervallo prendiamo gli estremi e il punto medio. Per questione di pura comodità, reindicizziamo tutti i nodi, in modo tale che:

$$x_0 = x_0 \quad x_1 = \hat{x}_1 \quad x_2 = x_1 \quad x_3 = \hat{x}_2 \quad x_4 = x_2 \quad \cdots$$

Questo per semplificare di molto la somma dell'integrale. Abbiamo allora che:

$$\mathcal{I}_{2,m} = \frac{H}{6}(\underbrace{f(x_0)+4f(x_1)+f(x_2)}_{} + \underbrace{f(x_2)+4f(x_3)+f(x_4)}_{} \cdots)$$

Notiamo che gli indici pari sono presi 2 volte, quelli dispari 4 volte e gli estremi una volta soltanto, quindi può essere riscritto come:

$$\mathcal{I}_{2,m} = \left(f(x_0) + 4 \sum_{k \text{ dispari}} f(x_k) + 2 \sum_{k \text{ pari}} f(x_k) + f(x_m) \right)$$

In Matlab, per prendere solo quelli pari o quelli dispari, si può sfruttare il passo due, cioè `a:2:b`
Il suo errore sarà quindi:

$$E_{2,m} = \frac{b-a}{180}\left(\frac{H}{2}\right)^4 f^{(4)}(\xi) \qquad \xi \in]a,b[$$

Osservazione. Come nel caso dell'interpolazione, dove possibile è meglio utilizzare le formule adattive o adattative, con la scelta dei nodi opportuna in base alla pendenza/variazione della funzione

5.6 Stima computazionale dell'ordine di Convergenza

Sia $E_{\cdot,m} = cH^p$. Vogliamo assicurarci che il valore p tenda al valore voluto (per esempio per la formula di Cavalieri-Simpson vorremmo che tendesse al valore voluto per $m \to \infty$). Dimostriamo

che è vera. Consideriamo (come avevamo fatto per l'interpolazione) due suddivisioni di $[a, b]$ tali che una abbia il doppio dei nodi dell'altra, allora abbiamo che:

$$E_{\cdot,m} = cH^p = c\left(\frac{b-a}{m}\right)^p \qquad \text{e} \qquad E_{\cdot,2m} = cH^p = c\left(\frac{b-a}{2m}\right)^p$$

Allora posso dividere il primo per il secondo e ottenere:

$$\frac{E_{\cdot,m}}{E_{\cdot,2m}} = \frac{c\left(\frac{b-a}{m}\right)^p}{c\left(\frac{b-a}{2m}\right)^p} = 2^p$$

Da cui, mettendo i logaritmi, abbiamo che:

$$\log\left(\frac{E_{\cdot,m}}{E_{\cdot,2m}}\right) = p\log(2) \qquad \Rightarrow \qquad p = \frac{1}{\log(2)}\log\left(\frac{C_{\cdot,m}}{E_{\cdot,2m}}\right) \xrightarrow{n\to+\infty} p \text{ teorico}$$

Esercizio (Metodo dei Coefficienti Indeterminati)**.** Data la formula di quadratura aperta (cioè estremi esclusi):

$$\int_{-2}^{2} f(x)dx \approx \alpha_1 f(-\sqrt{2}) + \alpha_2 f(0) + \alpha_3 f(\sqrt{3})$$

Determinare $\alpha_1, \alpha_2, \alpha_3$ in modo che la formula ridotta risulti di grado di precisione 2 (cioè esatta per polinomi di grado minore o uguale a 2)

Soluzione. Imponiamo l'esattezza della formula per polinomi di grado minore o uguale a 2.
Per $n = 0$ prendiamo $p_0(x) = 1$, per $n = 1$ prendiamo $p_1(x) = x$ e per $n = 2$ prendiamo $p_2(x) = x^2$.
Andando a sostituire otteniamo che:

$$f = p_0: \qquad \alpha_1 \cdot 1 + \alpha_2 \cdot 1 + \alpha_3 \cdot 1 = \int_{-2}^{2} 1dx = [x]_{-2}^{2} = 4$$

$$f = p_1: \qquad \alpha_1 \cdot (-\sqrt{2}) + \alpha_2 \cdot 0 + \alpha_3 \cdot (\sqrt{2}) = \int_{-2}^{2} xdx = \left[\frac{1}{2}x^2\right]_{-2}^{2} = 0$$

$$f = p_2: \qquad \alpha_1 \cdot 2 + \alpha_2 \cdot 0 + \alpha_3 \cdot 2 = \int_{-2}^{2} x^2dx = \left[\frac{1}{3}\right]_{-2}^{2} = \frac{16}{3}$$

Mettendo tutto insieme abbiamo che:

$$\begin{cases} \alpha_1 + \alpha_2 + \alpha 3 = 0 \\ -\sqrt{2}\alpha_1 + \sqrt{2}\alpha_3 = 0 \\ 2\alpha_1 + 2\alpha_3 = \frac{16}{3} \end{cases} \Rightarrow \begin{cases} \alpha_2 + \frac{8}{3} = 4 \Rightarrow \alpha_2 = \frac{4}{3} \\ \alpha_1 = \alpha_3 \Rightarrow \alpha_1 = \frac{4}{3} \\ 2\alpha_3 = \frac{8}{3} \Rightarrow \alpha_3 = \frac{4}{3} \end{cases}$$

■

Esercizio. Determinare $\alpha_0, \alpha_1, \beta_0, \beta_1$ in modo che la seguente formula sia esatta per polinomi di grado minore o uguale a 3:

$$\int_0^h f(x)dx \approx h(\alpha_0 f(0) + \alpha_1 f(h)) + h^2(\beta_0 f'(0) + \beta_1 f'(h))$$

5.7 Esercizi degli ultimi capitoli

Esercizio. Stimare l'integrale:

$$\mathcal{I} = \int_{-1}^{1} \frac{1}{2} e^{-x^2} dx$$

Mediante la formula dei trapezi e di Cavalieri-Simpson e stimare l'errore della formula dei trapezi

Soluzione. Abbiamo che la funzione in sé per sé è $f(x) = \frac{1}{2}e^{-x^2}$.
Con la formula dei trapezi abbiamo che:

$$\mathcal{I}_1 = \frac{b-a}{2}(f(a)+f(b)) = \frac{2}{2}(f(1)+f(-1)) = \frac{1}{2}(e^{-1}+e^{-1}) = e^{-1} = \frac{1}{e}$$

Con la formula di Cavalieri-Simpson abbiamo che:

$$\mathcal{I}_2 = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{b-a}{2}\right) + f(b)\right) = \frac{2}{6}\left(\frac{1}{2}e^{-1} + 4\frac{1}{2} + \frac{1}{2}e^{-1}\right) = \frac{1}{2}(e^{-1}+2)$$

I numeri possiamo lasciarli così, non c'è bisogno di fare calcoli superflui
Adesso possiamo calcolare l'errore:

$$|E_1| = \left|-\frac{(b-a)^3}{12}f''(\xi)\right| \leq \frac{8}{12}\|f''\|_\infty$$

Cerchiamo di stimare $\|f''\|_\infty$. Calcoliamo prima le derivate di f poi maggioriamo $\|f''\|_\infty$ con il massimo delle funzioni che la compongono:

$$f'(x) = -\frac{1}{2}2xe^{-x^2} \qquad f''(x) = -e^{-x^2} + 2x^2e^{-x^2} = e^{-x^2}(2x^2-1)$$

Abbiamo che, per $x \in [-1,1]$:

$$\max_{x\in[-1,1]} e^{-x^2} = 1 \qquad \max_{x\in[-1,1]} (2x^2-1) = 1$$

Quindi possiamo maggiorare:

$$\|f''\|_\infty \leq 1 \cdot 1 = 1$$

Da cui segue che:

$$|E_1| = \left|-\frac{(b-a)^3}{12}f''(\xi)\right| \leq \frac{8}{12}\|f''\|_\infty \leq \frac{2}{3}\cdot 1 = \frac{2}{3}$$

■

Esercizio. Stima il numero minimo di sottointervalli affinché il metodo di Cavalieri-Simpson composito dia un errore di alpiù $\varepsilon = 10^{-4}$ per l'approssimazione di:

$$\mathcal{I} = \int_0^\pi \left(e^{\frac{1}{2}x} + \cos x\right)$$

Soluzione. Notiamo che è richiesto solamente l'errore, quindi non è necessario trovare il valore di tale integrale. Anzi, risulterebbe in una perdita di tempo. Utilizziamo la formula per l'approssimazione dell'errore per la formula di Cavalieri-Simpson:

$$E_{2,m} = -\frac{b-a}{180}\left(\frac{H}{2}\right)^4 f^{(4)}(\xi) \qquad \text{con } H = \frac{b-a}{m}$$

Avevamo già preso sottointervalli divisi in quantità uguali. Facendo il valore assoluto di tale quantità abbiamo che:

$$|E_{2,m}| = \frac{\pi}{180}\cdot\frac{1}{16}\cdot\left(\frac{\pi}{4}\right)^4 \cdot f^{(4)}(\xi)$$

Anche qui, non c'è il minimo bisogno di fare tutti tutti i conti. Andiamo, come prima, a calcolare tutte le derivate fino alla quarta:

$$f'(x) = \frac{1}{2}e^{\frac{1}{2}x} - \sin x \quad f''(x) = \frac{1}{4}e^{\frac{1}{2}x} - \cos x \quad f'''(x) = \frac{1}{8}e^{\frac{1}{2}x} + \sin x \quad f^{(4)}(x) = \frac{1}{16}e^{\frac{1}{2}x} + \cos x$$

Poi cerchiamo di maggiorare $\|f^{(4)}\|_\infty$ nuovamente prendendo il massimo delle due funzioni che costituiscono la funzione f:

$$\|f^{(4)}\|_\infty \leq \max_{x\in[0,\pi]} \frac{1}{16}e^{\frac{1}{2}x} + \max_{x\in[0,\pi]} \cos x = \frac{1}{16}e^{\frac{\pi}{2}} + 1$$

Adesso imponiamo il fatto che l'errore sia minore di ε:

$$|E_{2,m}| = \frac{\pi}{180}\cdot\frac{1}{16}\cdot\left(\frac{\pi}{4}\right)^4 \cdot f^{(4)}(\xi) \leq \frac{\pi^5}{16\cdot 180}\frac{1}{m^4}\left(\frac{1}{16}e^{\frac{\pi}{2}} + 1\right) < \varepsilon \equiv 10^{-4}$$

Ricaviamo ora il numero minimo di sotto intervalli, risolvendo per m dalla disuguaglianza sopra. In questo modo abbiamo che:

$$m > \sqrt[4]{10^4 \cdot \frac{\pi^5}{16\cdot 180}\left(\frac{1}{16}e^{\frac{\pi}{2}} + 1\right)}$$

Cioè il numero minimo di intervalli è:

$$m \geq \left\lceil \sqrt[4]{\frac{\pi^5}{16\cdot 180}\left(\frac{1}{16}e^{\frac{\pi}{2}} + 1\right)} \right\rceil$$

■

Esercizio. Sia $h > 0$ e sia $\mathcal{I}(f,h)$ definito come:

$$\mathcal{I}(f,h) = \int_0^h f(x)dx$$

1. Calcolare $\mathcal{I}(f,h)$ e $\mathcal{I}_1(f,h)$ (cioè con la formula dei trapezi) e calcolare l'errore vero $\mathcal{I}(f,h) - \mathcal{I}_1(f,h)$ per $f(x) = x^2 + x^{5/2}$
2. Ripeti per $f(x) = x^2 + \sqrt{x}$. Valuta la differenza dell'errore per $h \to 0$, valutandone l'ordine di grandezza

Soluzione. 1 Sia $f(x) = x^2 + x^{5/2}$, allora abbiamo che:

$$\mathcal{I}(f,h) = \int_0^h x^2 + x^{5/2} dx = \left[\frac{1}{3}x^3 + \frac{2}{7}x^{7/2}\right]_0^h = \frac{1}{3}h^3 + \frac{2}{7}h^{7/2}$$

$$\mathcal{I}_1(f,h) = \frac{b-a}{2}(f(a)+f(b)) = \frac{h}{2}(f(0)+f(h)) = \frac{h}{2}(0 + h^2 + h^{5/2}) = \frac{1}{2}h^3 + \frac{1}{2}h^{7/2}$$

Allora abbiamo che l'errore è:

$$\mathcal{I}(f,h) - \mathcal{I}_1(f,h) = \left(\frac{1}{3} - \frac{1}{2}\right)h^3 + \left(\frac{2}{7} - \frac{1}{2}\right)h^{7/2} = -\frac{1}{6}h^3 - \frac{3}{14}h^{7/2} = h^3\left(-\frac{1}{6} - \frac{3}{14}h^{\frac{1}{2}}\right)$$

Quindi abbiamo che per $h \to 0$, abbiamo che l'errore scendo come h^3

2 Sia adesso $f(x) = x^2 + \sqrt{x}$. Allora abbiamo che:

$$\mathcal{I}(f,h) = \int_0^h x^2 + \sqrt{x} dx = \frac{1}{3}h^3 + \frac{2}{3}h^{3/2}$$

$$\mathcal{I}_1(f,h) = \frac{h}{2}(f(0)+f(h)) = \frac{h}{2}(h^2 + \sqrt{h}) = \frac{1}{2}h^3 + \frac{1}{2}h^{3/2}$$

Da cui segue direttamente che l'errore è:

$$\mathcal{I}(f,h) - \mathcal{I}_1(f,h) = \left(\frac{1}{3} - \frac{1}{2}\right)h^2 + \left(\frac{2}{3} - \frac{1}{2}\right)h^{3/2} = -\frac{1}{6}h^3 + \frac{1}{6}h^{3/2} = \frac{1}{6}h^{3/2}(1 - h^{3/2})$$

Quindi per $h \to 0$ ho che l'errore scende come $h^{3/2}$. Rispetto a prima troviamo un errore minore rispetto a prima e rispetto a quanto il teorema ci dice. *Perché?* In questa funzione viene a mancare l'ipotesi di C^2, infatti non è neanche C^1. Quindi non potevo aspettarmi un qualcosa del tipo $h^3 \equiv (b-a)^3$. L'errore scendo più lentamente rispetto a prima proprio per questo motivo ■

Esercizio (Esame 8/1/2021)**.** È data la funzione $F(x) : [\frac{1}{2}, 1] \to \mathbb{R}$ definita come:

$$F(x) = \int_0^x \frac{2t^2 - 3t + 1}{t^2 + 1} dt$$

- Per $x \in [\frac{1}{2}, 1]$ determinare la formula di quadratura composita dei rettangoli per approssimare $F(x)$
- Sfruttando il primo punto, usando solo 2 sottointervalli $m = 2$, proponi una procedura per l'approssimazione di F nell'intervallo $[\frac{1}{2}, 1]$ mediante interpolazione con polinomi di grado 2

Soluzione. Per questioni di comodità poniamo:

$$g(t) = \frac{2t^2 - 3t + 1}{t^2 + 1}$$

1 Per $x \in [\frac{1}{2}, 1]$, abbiamo che:

$$F(x) = \mathcal{I}_{0,m} = H(g(\hat{x}_1) + g(\hat{x}_2) + \cdots + g(\hat{x}_m))$$

Dove $\hat{x}_i$ sono i punti medi di ogni intervallo. Nel nostro caso abbiamo $a = 0$ e $b = x$, quindi la lunghezza dell'intervallo è variabile:

$$H = \frac{b-a}{m} = \frac{x-0}{m} = \frac{x}{m}$$

Notiamo però che possiamo scrivere i vari punti medi in un altro modo:

$$\hat{x}_1 = \frac{H}{2} + a = \frac{H}{2} \qquad \hat{x}_2 = \frac{H}{2} + H + a = \frac{3}{2}H$$

Quindi possiamo ricavare la formula di quadratura come:

$$\mathcal{I}_{0,m} = \frac{x}{m}\left(g\left(\frac{H}{2}\right) + g\left(\frac{3}{2}H\right) + \cdots + g\left(x - \frac{H}{2}\right)\right)$$

2 Abbiamo $F : [\frac{1}{2}, 1] \to \mathbb{R}$ e vogliamo approssimarlo con polinomi di grado 2. Nel fare ciò, facciamo finta di non avere un integrale, ma di avere una funzione qualsiasi. In particolare, approssimiamo la funzione con polinomi di secondo grado. Possiamo prendere tre nodi equidistanti. In particolare possiamo prendere:

$$x_0 = a = \frac{1}{2} \qquad x_1 = \frac{b-a}{2} = \frac{3}{4} \qquad x_2 = b = 1$$

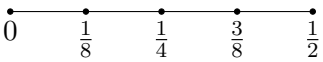
In questo modo possiamo utilizzare i polinomi di Lagrange per approssimare la funzione. In particolare:

$$p_2(x) = y_0 L_0(x) + y_1 L_1(x) + y_2 L_2(x)$$

Il problema adesso sta nell'effettivo trovare y_0, y_1, y_2. Nel fare ciò possiamo approssimare l'integrale, infatti:

$$y_0 = F(x_0) = \int_0^{\frac{1}{2}} g(t)dt \qquad y_1 = F(x_1) = \int_0^{\frac{3}{4}} g(t)dt \qquad y_2 = F(x_2) = \int_0^{1} g(t)dt$$

Il testo dell'esercizio ci impone di approssimare i vari integrali utilizzando la formula di quadratura dell'esercizio precedente e con soli due intervalli.
Facciamolo per bene per il primo, gli altri due è analogo. Il primo intervallo è $[0, \frac{1}{2}]$. Visto che dobbiamo prendere due intervalli, possiamo prendere il punto medio $\frac{1}{4}$, in modo da avere due intervalli di stessa lunghezza $[0, \frac{1}{4}]$ e $[\frac{1}{4}, \frac{1}{2}]$. Visto che dobbiamo sostanzialmente applicare la formula dei trapezi su questi intervalli, dobbiamo prendere nuovamente i punti medi, $\frac{1}{8}$ per il primo e $\frac{3}{8}$ per il secondo.



Applichiamo quindi la formula di prima sui due intervalli. Quindi abbiamo che:

$$F(x_0) = \int_0^{\frac{1}{2}} g(t)dt \approx H(g(\hat{x}_1) + g(\hat{x}_2)) = \frac{1}{4}\left(g\left(\frac{1}{8}\right) + g\left(\frac{3}{8}\right)\right) \equiv \hat{F}(x_0)$$

Poi ci sarebbero da fare conti ma non sono rilevanti
Vediamo per $F(x_1)$ e $F(x_1)$.

Per $F(x_1)$ abbiamo che gli estremi sono 0 e $\frac{3}{4}$, il punto medio e lunghezza dei sottointervalli H è $\frac{3}{8}$ e i punti medi dei rispettivi sottointervalli sono $\frac{3}{16}$ e $\frac{9}{16}$. Da questo segue che:

$$F_1(x) = \int_0^{\frac{3}{4}} g(t)dt \approx H(g(\hat{x}_1) + g(\hat{x}_2)) = \frac{3}{8}\left(g\left(\frac{3}{16}\right) + g\left(\frac{9}{16}\right)\right) \equiv \hat{F}(x_1)$$

Per $F(x_2)$ abbiamo che gli estremi sono 0 e 1, il punto medio e lunghezza dei sottointervalli è $H + \frac{1}{2}$ e i rispettivi punti medi sono $\frac{1}{4}$ e $\frac{3}{4}$. Quindi abbiamo che:

$$F(x_2)) \int_0^1 g(t)dt \approx \frac{1}{2}\left(g\left(\frac{1}{4}\right) + g\left(\frac{3}{4}\right)\right) \equiv \hat{F}(x_2)$$

Non serve andare avanti con i conti. Abbiamo quindi che il nostro polinomio è:

$$p_2(x) = \hat{F}(x_0)L_0(x) + \hat{F}(x_1)L_1(x) + \hat{F}(x_2)L_2(x)$$

Dove abbiamo che:

$$L_0(x) = \frac{(x-\frac{3}{4})(x-1)}{(\frac{1}{2}-\frac{3}{4})(\frac{1}{2}-1)} \qquad L_1(x) = \frac{(x-\frac{1}{2})(x-1)}{(\frac{3}{4}-\frac{1}{2})(\frac{3}{4}-1)} \qquad L_2(x) = \frac{(x-\frac{1}{2})(x-\frac{3}{4})}{(1-\frac{1}{2})(1-\frac{3}{4})}$$

■

Esercizio (Esame 17/12/2019)**.** È data la funzione $f(x) = \sin(\pi x)$ ed i nodi $x_1 = \frac{1}{4}$ e $x_2 = \frac{1}{2}$. Determinare il polinomio di grado 3 tale che:

$$p(x_i) = f(x_i) \qquad p'(x_i) = f'(x_i) \qquad i \in \{1, 2\}$$

Commentare poi sull'esistenza e sull'unicità di tale polinomio

Soluzione. Sapendo che $f(x) = \sin(\pi x)$, possiamo subito calcolare la derivata come:

$$f'(x) = \pi\cos(\pi x)$$

Sapendo che vogliamo un polinomio di terzo grado, possiamo già porre:

$$p(x) = a_0 + a_1x + a_2x^2 + a_3x^3 \qquad p'(x) = a_1 + 2a_2x + 3a_3x^2$$

Iniziamo ad imporre tutte le condizioni. Allora abbiamo che:

$$\begin{cases} p(x_1) = f(x_1) \\ p(x_2) = f(x_2) \\ p'(x_1) = f'(x_1) \\ p'(x_2) = f'(x_2) \end{cases} \Rightarrow \begin{cases} a_0 + \frac{1}{4}a_1 + \frac{1}{16}a_2 + \frac{1}{64}a_3 = \sin(\frac{\pi}{4}) = \frac{\sqrt{2}}{2} \\ a_0 + \frac{1}{2}a_1 + \frac{1}{4}a_2 + \frac{1}{8}a_3 = \sin(\frac{\pi}{2}) = 1 \\ a_1 + \frac{2}{4}a_2 + \frac{3}{16}a_3 = \pi\cos(\frac{\pi}{4}) = \frac{\sqrt{2}}{2} \\ a_1 + \frac{2}{2}a_2 + \frac{3}{4}a_3 = \pi\cos(\frac{\pi}{2}) = 0 \end{cases}$$

Notiamo che questo è un sistema lineare, quindi possiamo scriverlo come:

$$\begin{pmatrix} 1 & \frac{1}{4} & \frac{1}{16} & \frac{1}{64} \\ 1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} \\ 0 & 1 & \frac{1}{2} & \frac{3}{16} \\ 0 & 1 & 1 & \frac{3}{4} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} \\ 1 \\ \frac{\sqrt{2}}{2} \\ 0 \end{pmatrix}$$

Da qui ottengo i coefficienti del polinomio che stavo cercando se e solo se la matrice è non singolare. Per quanto fatto nel modulo precedente, se tale soluzione esiste, è anche unica. Quindi esiste ed è unico il polinomio. (Andrebbe effettivamente verificato che la matrice sia non singolare.) ■

Esercizio. È data la funzione $f(x) = \log(2+x)$ con $x \in [-1,1]$ ed il polinomio interpolante p_n di grado n nei nodi di Chebyschev in $[-1,1]$.

- Ottieni una stima per $\|f - p_n\|_\infty$
- Confronta la stima con una stima opportuna per $\|f - t_n\|_\infty$ dove t_n è il polinomio di Taylor di grado n di f intorno all'origine

Soluzione. 1 In generale abbiamo che:

$$\|f - p_n\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}\|\omega\|_\infty$$

Nel caso dei nodi di Chebyschev abbiamo che $\|\omega\|_\infty$ è minimizzato e vale:

$$\|\omega\|_\infty = \frac{1}{2^n}$$

Da cui segue che la stima per

$$\|f - p_n\|_\infty$$

è:

$$\|f - p_n\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}\|\omega\|_\infty = \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}\frac{1}{2^n}$$

2 Abbiamo che il polinomio di Taylor di grado n di f centrato in 0 è:

$$f(x) = f(0) + f'(0)x + \frac{f''(0)}{2}x^2 + \cdots + \frac{f^{(n)}(0)}{n!} + \frac{f^{(n+1)}(\xi_x)}{(n+1)!}x^{n+1}$$

Segue quindi che:

$$\|f - t_n\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}\|x^{n+1}\|_\infty = \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}$$

Quindi la stima con Chebyschev è migliore rispetto a quella con Taylor, $\forall n \geq 1$ ■

Esercizio. Sono dai i nodi $x_0 = -1$, $x_1 = 0$ e $x_2 = 1$ e la funzione f definita come:

$$f(x) = x^2 \sin\left(\frac{\pi}{2}x\right)$$

1. Calcolare il polinomio di grado $n = 2$ interpolante f sui nodi nella forma:

   $$p(x) = \alpha_2 x^2 + \alpha_1 x + \alpha_0$$

   Mediante il metodo di Vandermonde
2. Calcolare il polinomio di grado $n = 1$ che minimizza la distanza di interpolazione (nel senso dei minimi quadrati) rispetto ad f, usando i nodi dati.
3. Supponendo $x_2 = x_1 + \varepsilon$ con $\varepsilon \in \mathbb{R}^+$, commenta sul condizionamento del problema al punto 1 ed in particolare sulla possibile quasi singolarità della matrice.

Soluzione. 1 Costruiamo la matrice di Vandermonde con i nodi e poniamo il sistema lineare con i valori della funzione:

$$\begin{pmatrix} 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

Per la soluzione basta quindi risolvere il sistema lineare

2 Sia quindi il polinomio di grado 1:

$$p_1(x) = \alpha_0 + \alpha_1 x$$

Poniamo la condizione di interpolazione:

$$p_1(x_i) = f(x_i) \qquad \Rightarrow \qquad \begin{pmatrix} 1 & -1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} \qquad \Leftrightarrow \qquad V\underline{\alpha} = \underline{f}$$

Possiamo quindi applicare il metodo dei minimi quadrati con la matrice V di dimensione 3×2.

$$\min_{\underline{\alpha}} \|\underline{f} - V\underline{\alpha}\| \qquad \Rightarrow \qquad V^T V\underline{\alpha} = V^T \underline{f}$$

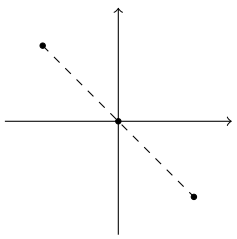
Abbiamo quindi che:

$$V^T V = \begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 0 \\ 0 & 2 \end{pmatrix} \qquad V^T \underline{f} = \begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 0 \\ -2 \end{pmatrix}$$

Per cui abbiamo che:

$$\begin{pmatrix} 3 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \begin{pmatrix} 0 \\ -2 \end{pmatrix} \qquad \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \begin{pmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 0 \\ -2 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$

Se volessimo rappresentare la soluzione ottenuta avremmo che:



3 Riprendiamo la matrice e sostituiamo il terzo nodo con $x_2 = x_1 + \varepsilon = 0 + \varepsilon = \varepsilon$:

$$V = \begin{pmatrix} 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & \varepsilon & \varepsilon^2 \end{pmatrix}$$

Per studiare la singolarità di questa matrice possiamo studiare la singolarità di un'altra matrice simile a V:

$$V' = \begin{pmatrix} 1 & 0 & 0 \\ 1 & -1 & 1 \\ 1 & \varepsilon & \varepsilon^2 \end{pmatrix}$$

Per cui V' è singolare solo se lo è anche il blocco in blu. Andiamo quindi a studiarne il determinante:

$$\det \begin{pmatrix} -1 & 1 \\ \varepsilon & \varepsilon^2 \end{pmatrix} = -\varepsilon^2 - \varepsilon = -\varepsilon(\varepsilon + 1) \to 0 \quad \text{per } \varepsilon \to 0$$

■

Esercizio (Esame 19/7/2022)**.** Sia $f(x) = \sqrt{(1-x^2)^3}$ per $x \in [-1, 1]$:

1. Determinare il polinomio p_2 interpolante f in modo esplicito
2. Determinare il polinomio p_2 con i nodi di Chebyschev
3. Per $n > 2$ analizzare una stima dei punti precedenti per $n \to +\infty$

Soluzione. 1 Come negli esercizi precedenti con $x_0 = -1$, $x_1 = 0$ e $x_2 = 1$, da cui:

$$p_2(x) = \alpha_2 x^2 + \alpha_1 x + \alpha_0$$

2 In maniera analoga con i nodi di Chebyschev.
3 Sappiamo che $f \notin C^\infty$, in particolare $f \notin C^2$ e

$$|f(x) - p_n(x)| = \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega(x)$$

Da questa non possiamo discutere l'errore per $n \to +\infty$ per gli equispaziati. Per Chebyschev invece, abbiamo la convergenza uniforme per Bernsetein ■

5.8 Formule di Newton - Cotes

Volendo è possibile generalizzare le formule di quadratura anche per gradi qualsiasi di polinomi. Infatti, più in generale abbiamo che:

$$p_n(x) = \sum_{i=0}^{n} f(x_i)L_i(x) \qquad \text{per } x_k = x_0 + hk \quad k \in \{1, ...\}$$

Per cui abbiamo che:

$$\mathcal{I}_n = \int_a^b \sum_{i=0}^{n} f(x_i)L_i(x)dx = \sum_{i=0}^{n} f(x_i) \int_a^b L_i(x)dx = \sum_{i=0}^{n} f(x_i)\alpha_i$$

Dove abbiamo posto che:

$$\alpha_i = h\omega_i$$

Con ω_i valori che possono essere registrati in tabelle. Infatti, facendo i conti, abbiamo che:

$$\int_a^b L_i(x)dx = \int_a^b \prod_{k=0, k\neq i}^{n} \frac{(x - x_k)}{(x_i - x_k)}dx$$

Sapendo che i nodi sono equispaziati abbiamo che $x_k = x_0 + kh$ con $h = |x_{i+1} - x_i|$. Inoltre un qualsiasi punto $x \in [a, b]$ può essere scritto come:

$$x = x_0 + ph \qquad p \in [0, n]$$

Per cui possiamo scrivere i vari termini della produttoria possono essere scritti come:

$$\frac{(x-x_k)}{(x_i-x_k)} = \frac{(p-k)h}{(i-k)h} = \frac{p-k}{i-k}$$

Qui k e i sono due indici.
Riprendendo il fatto che possiamo scrivere:

$$x = x_0 + ht \qquad \Rightarrow \qquad dx = hdt$$

Per cui, andando a sostituire abbiamo che:

$$\int_a^b \prod_{k=0,k\neq i}^{n} \frac{(x-x_k)}{(x_i-x_k)}dx = \int_0^n \prod_{k=0,k\neq i}^{n} \frac{t-k}{i-k}dt = \omega_i$$

Quindi dipende esclusivamente dalla scelta dei nodi (se sono equispaziati), quindi posso scegliere degli ω_i.

Vediamone l'accuratezza.
Per n pari e per $f \in C^{(n+2)}([a,b])$ abbiamo che:

$$E_n(f) = \frac{M_n}{(n+2)!}h^{n+3}f^{(n+2)}(\xi) \qquad \xi \in (a,b)$$

M_n è una costante che dipende dai nodi.
Per n dispari e per $f \in C^{(n+1)}([a,b])$ abbiamo che:

$$E_n(f) = \frac{N_n}{(n+1)!}h^{n+2}f^{(n+1)}(\xi) \qquad \xi \in (a,b)$$

Per aumentare accuratezza non è saggio aumentare il numero di nodi n, meglio sfruttare le composite. Per scegliere il numero adatto di intervalli possiamo risolvere:

$$|cH^k f^{(k)}(\xi)| \leq |c|H^k \|f^{(k)}\|_\infty < \varepsilon$$

Quindi:

$$m \geq \left(\frac{|c|\|f^{(k)}\|_\infty}{\varepsilon}(b-a)^k\right)^{1/k}$$

6 Approssimazione di Derivate (differenze finite)

6.1 Somme in Avanti e in Indietro

Il problema che ci prefiggiamo di risolvere in questa sezione è: *data una funzione* $f : I \to \mathbb{R}$ *e dato* $x_i \in I$*, supponendo che esista* $f'(x_i)$ *definito come:*

$$f'(x) = \lim_{h \to 0} \frac{f(x_i + h) - f(x_i)}{h}$$

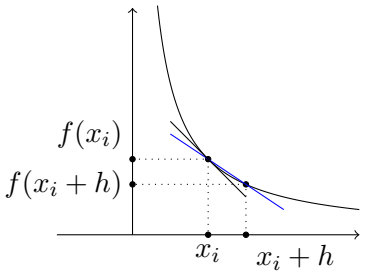
Vogliamo approssimare $f'(x)$ *mediante il rapporto incrementale.*

Definizione 6.1.1: Differenze Finite in Avanti

Definiamo le **Differenze Finite in Avanti** come:

$$f_i^{FD} = \frac{f(x_i + h) - f(x_i)}{h} \qquad h > 0$$

FD = Forward differences. Quindi in questo caso abbiamo che $x_i + h$ sta sempre dopo x_i.



Valutiamone l'errore (con una funzione f sufficientemente regolare). Con Taylor abbiamo che:

$$f(x_i + h) = f(x_i) + hf'(x_i) + \frac{1}{2}h^2 f''(\xi_i) \qquad \xi_i \in (x_i, x_i + h)$$

Per cui abbiamo che:

$$f'(x_i) = \frac{f(x_i + h) - f(x_i)}{h} - \frac{1}{2}hf''(\xi_i) \qquad \text{per } h \to +\infty$$

L'errore quindi va a 0 come h^1, quindi l'errore è un $\mathcal{O}(h^1)$

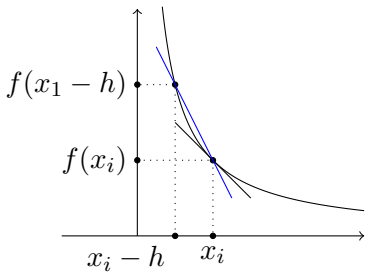
Possiamo fare una cosa simile anche con le somme in indietro:

Definizione 6.1.2: Differenze Finite all'Indietro

Definiamo le **Differenze Finite all'Indietro** come:

$$f_i^{BD} = \frac{f(x_i) - f(x_i - h)}{h} \qquad h \geq 0$$

Qui abbiamo che BD sta per Backward differences. Graficamente abbiamo che:



Abbiamo quindi che l'approssimazione è analoga a quanto fatta in precedenza:

$$f(x_i - h) = f(x_i) - hf'(x_i) + \frac{1}{2}h^2 f''(\hat{\xi}_i) \qquad \text{per } \hat{\xi}_i \in (x_i - h, x_i)$$

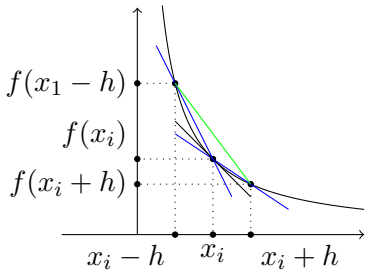
Da cui segue che:

$$f'(x_i) = \frac{f(x_i) - f(x_i - h)}{h} + \frac{1}{2}hf''(\hat{\xi}_i)$$

Per cui abbiamo sempre che l'errore decresce come $\mathcal{O}(h^1)$

Se invece prendiamo come estremi $x_i - h$ e $x_i + h$ abbiamo un risultato migliore. Infatti, sviluppando con Taylor fino al terzo ordine, abbiamo che, per $h \to +\infty$:

$$f(x_i + h) = f(x_i) + hf'(x_i) + \frac{1}{2}h^2 f''(x) + \frac{1}{6}h^3 f'''(\xi_i)$$
$$f(x_i - h) = f(x_i) - hf'(x_i) + \frac{1}{2}h^2 f''(x) - \frac{1}{6}h^3 f'''(\xi_i)$$



Andiamo a sottrarre termine a termine:

$$f(x_i + h) - f(x_1 - h) = 2hf'(x_i) + \frac{1}{6}h^2 f'''(\xi_i) + \frac{1}{6}h^3 f'''(\hat{x}_i) \qquad h \to 0$$

Ponendo gli ultimi addendi come $\mathcal{O}(h^3)$, abbiamo che:

$$f'(x_i) = \frac{f(x_i + h) - f(x_i - h)}{2h} + \mathcal{O}(h^2)$$

Questa è un'approssimazione di ordine 2, quindi l'errore va a 0 come h^2

Tutto questo risulta comodo nel caso in cui dobbiamo avere la derivata in qualche punto. Il metodo sfruttando come punti $x_i - h$ e $x_i + h$ non può essere sempre usato, come nel caso delle spline, in cui stiamo cercando la derivata in un estremo dell'intervallo.

6.2 Approssimazione di Derivate Seconde

In questo caso, rispetto a prima, ci basterà aumentare di un grado lo sviluppo di Taylor:

$$f(x_i + h) = f(x_i) + hf'(x_i) + \frac{1}{2}h^2 f''(x) + \frac{1}{6}h^3 f'''(\xi_i) + \frac{1}{4!}h^4 f^{(4)}(\xi_i)$$
$$f(x_i - h) = f(x_i) - hf'(x_i) + \frac{1}{2}h^2 f''(x) - \frac{1}{6}h^3 f'''(\xi_i) + \frac{1}{4!}h^4 f^{(4)}(\xi_i)$$

Sommando i due termini abbiamo che:

$$f(x_i + h) + f(x_i - h) = 2f(x_i) + h^2 f''(x_i) + \frac{1}{2}\frac{1}{4!}h^4 f^{(4)}(\tilde{\xi}_i) \qquad \tilde{\xi}_i \in (x_i - h, x_i + h)$$

Dividendo poi per h^2 e riordinando i termini abbiamo che:

$$f''(x_i) = \frac{(f(x_i - h) - 2f(x_i) + f(x_i + h))}{h^2} - \frac{1}{48}h^2 f^{(4)}(\tilde{\xi}_i)$$

Da cui segue che:

$$f''(x_i) - \frac{f(x_i - h) - 2f(x_i) + f(x_i + h)}{h^2} = \mathcal{O}(h^2)$$

Per cui l'errore va a 0 come h^2.

Osservazione. Per il calcolo della derivata seconda bisogna fare una valutazione della funzione tre volte

Questo metodo di approssimazione può essere utilizzato per l'equazione del moto, nel quale si approssima $\ddot{u} = f$ con $f = f(t), t \in [0, T]$, oppure per una diffusione di qualche tipo, con $u'' = f$, $f = f(x)$ e $x \in [a, b]$

7 Equazioni non Lineari

7.1 Presentazione del Problema

Data una funzione $f : (a, b) \to \mathbb{R}$ vogliamo determinare un'approssimazione $\tilde{x}$ a x^* zero di f in (a, b), cioè $f(x^*) = 0$

Facciamo un paio di osservazioni

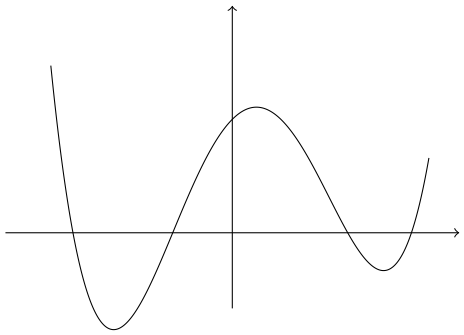
Osservazione. Questo problema può risultare complesso se teniamo conto del fatto che:

- Non è detto che esista sempre x^*
- Se esiste, non è detto che sia unico (in quanto c'è la possibilità che troviamo lo zero sbagliato se ce ne sono più di 1)
- Se lo troviamo, non è detto che lo troviamo in forma chiusa, cioè x^* =qualcosa
- Se lo troviamo in forma chiusa, non è detto che sia calcolabile o approssimabile

Il problema è quindi numericamente difficile e la risoluzione ne genera molto interesse in quanto molti modelli ne fatto utilizzo. Si tratta quindi di un modello molto attuale.

In questo corso ci limiteremo ad un caso scalare, ma volendo può essere apportato ad un caso più generale in ambito vettoriale in $\mathbb{R}^n$. Tuttavia in questo caso ci sono dei vincoli che per il momento è meglio lasciar perdere.

Osservazione. La conoscenza del grafico della funzione f può essere di gran aiuto, in quanto è in grado di darci delle prime informazioni:



Osservazione. Se riusciamo a scrivere una funzione come differenza di due funzioni, possiamo allora porre un sistema:

$$f(x) = \varphi_1(x) - \varphi_2(x) = 0 \qquad \begin{cases} y_1 = \varphi_1(x) \\ y_2 = \varphi_2(x) \end{cases}$$

In questo modo la loro intersezione è esattamente lo zero della funzione

Scriviamo un procedimento generale:
Dato $x_0 \in [a, b]$, vogliamo determinare una successione $\{x_k\}_{k\geq 0}$ tale che sotto determinate ipotesi o condizioni si abbia che:

$$x_k \xrightarrow{k\to+\infty} x^*$$

Questo è un metodo iterativo come quelli visti nel semestre scorso. In generale, in questo ambito ne vedremo molti iterativi, non ce ne sono molti diretti

Definizione 7.1.1: Convergenza Lineare

Una successione $\{x_k\}_{k\geq 0}$ si dice che **Converge Linearmente** a x^* se:

$$\exists c \in [0,1] : \lim_{k\to+\infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|} = c$$

È vero che è una condizione al limite, ma si spera che effettivamente decresca tale quantità

Definizione 7.1.2: Convergenza di Ordine $p > 1$

Si dice che una successione $\{x_k\}_{k\geq 0}$ **Converge con ordine** $p > 1$ a x^* se:

$$\exists C > 0 : \lim_{k\to+\infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^p} = C$$

In questo caso la quantità C prende il nome di **Fattore Asintotico di Convergenza**

Il nostro obiettivo sarà quindi quello di studiare metodi che abbiano una convergenza più veloce (quindi un valore C più piccolo e un p più grande)

Osservazione. La convergenza può dipendere dal dato iniziale x_0 (infatti se è troppo lontana può convergere ad un altro punto, diverso da quello desiderato, oppure può capitare che non converga e basta). In tal caso si stratta di **Convergenza Locale**. Vedremo per esempio come con Newton il punto dovrà essere necessariamente vicino.
Se la convergenza invece non dipende da x_0, allora si tratta di **Convergenza Globale**

7.2 Condizionamento del Problema

Volendo possiamo scrivere la condizione $f(x) = 0$ come:

$$\varphi(x) - d = 0$$

Dove φ non è altro che una funzione che differisce di una costante d dalla funzione f stessa. Allora potremo scrivere che:

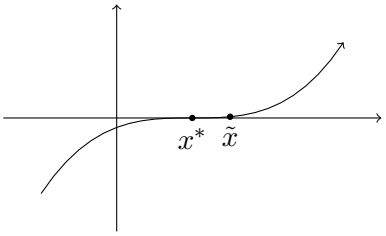
$$x = \varphi^{-1}(d)$$

Notiamo che questo è quanto avevamo fatto ad inizio semestre, ponendo $x = F(d)$, in particolare possiamo definire con:

$$C_{ABS} = F'(d) = (\varphi^{-1})'(d) = \frac{1}{\varphi'(x^*)} = \frac{1}{f'(x^*)}$$

Dove avevamo che $f(x^*) = 0$

Stiamo quindi studiando quanto le perturbazioni modificando la funzione stessa:infatti se abbiamo una funzione del genere:



Qui abbiamo che $f(\tilde{x}) \approx 0$, in quanto abbiamo che la derivata prima è molto piccola in quel punto. Quindi siamo vicini a zero. In questo modo la possibile di confondere, o di sbagliare, ad individuare lo zero è molto alta.

In tutti questi discorsi stiamo supponendo che x^* sia semplice, in modo tale che $f'(x^*) \neq 0$. Altrimenti tutto questo non varrebbe. Infatti, se la sua molteplicità m fosse maggiore di 1, allora avremmo che:

$$f^{(k)}(x^*) = 0 \qquad \forall k < m$$

In maniera del tutto analoga per φ. Sviluppando per x^* e ponendo $\delta x = x - x^*$ otteniamo che:

$$d + \delta d = \varphi(x^*) + \varphi'(x^*)(x - x^*) + \cdots + \frac{\varphi^{(m-1)}(x - x^*)}{(m-1)!}(x - x^*)^{m-1} + \frac{\varphi^{(m)}(x^*)}{m!}$$

$$d + \delta d = \underbrace{\varphi(x^*)}_{d} + \frac{\varphi^{(m)}(x^*)}{m!}(\delta x)^m \qquad \Rightarrow \qquad \delta d = \frac{\varphi^{(m)}(x^*)}{m!}$$

Da cui segue che:

$$\delta x = \left(\frac{\delta d \cdot m!}{\varphi^{(m)}(x^*)}\right)^{\frac{1}{m}}$$

7.3 Metodo di Bisezione

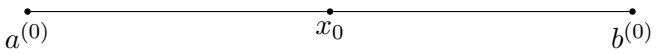
Iniziamo questo paragrafo ricordando un teorema di Analisi Matematica $1A$:

Teorema 7.3.1: degli Zeri

Sia $f \in X([a,b])$. Supponiamo $f(a)f(b) < 0$, allora esiste $x^* \in (a,b)$ tale che $f(x^*) = 0$

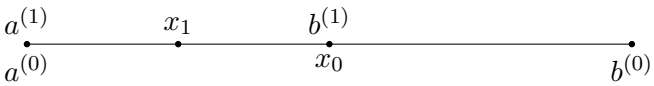
Considerazioni Aggiuntive. *Questo teorema ci dice già che come fare per trovare lo zero della funzione e ci da un metodo che funziona sempre.*

Andiamo a vedere come funziona:
Definiamo con $a^{(0)} = a$, $b^{(0)} = b$ e $x_0 = \frac{a+b}{2}$ il loro punto medio.

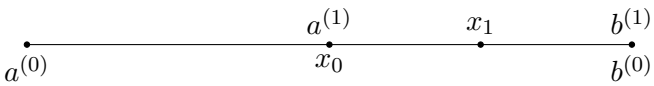


Allora possiamo definire la ricorrenza cercata nel seguente modo:

- Se $f(a^{(k)})f(x_k) < 0$, allora possiamo porre $a^{(k+1)} = a^{(k)}$ e $b^{(k+1)} = x_k$, in questo modo ci stiamo spostando verso sinistra:



- Altrimenti abbiamo che $f(n^{(k)})f(x_k) < 0$, quindi definiamo $a^{(k+1)} = x_k$ e $b^{(k+1)} = b^{(k)}$



In entrambi i casi, definiamo il punto medio come:

$$x_{k+1} = \frac{a^{(k+1)} + b^{(k+1)}}{2}$$

Proseguendo in questo modo, abbiamo sicuramente convergenza. In particolare, ponendo con $I = [a^{(k)}, b^{(k)}]$ e con $|I_k| = b^{(k)} - a^{(k)}$, allora, se riprendiamo la definizione di errore che avevamo dato $e_k = x^* - x_k$, abbiamo che:

$$|e_k| \leq \frac{|I_k|}{2} \leq \frac{b-a}{2^{k+1}} \xrightarrow{k \to +\infty} 0$$

Quindi c'è una convergenza globale rispetto a quanto fatto prima.

Osservazione. Visto che si conosce a priori $b - a$, come faccio a dire indicativamente quante iterazioni ci vogliono? In particolare (per esempio) quante iterazioni ci voglio per avere un errore minore di 10^{-4}

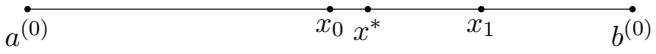
$$\frac{b-a}{2^{k+1}} \leq 10^{-4} \quad \Rightarrow \quad 2^{k+1} \geq \frac{b-a}{10^{-4}} \quad \Rightarrow \quad (k+1)\ln 2 \geq \ln\left(\frac{b-a}{10^{-4}}\right)$$

Quindi abbiamo che:

$$k + 1 \geq \frac{1}{\ln 2} \ln\left(\frac{b-a}{10^{-4}} = \overline{k}\right)$$

Qui abbiamo quindi la possibilità di sapere quante iterazioni sono necessarie al fine di avere un errore prescelto. Saranno necessarie al più $\overline{k}$ iterazioni.

Osservazione. Notiamo che non c'è convergenza monotona, in particolare non è un metodo di convergenza di ordine 1, in quanto converge molto lentamente.
Non si può parlare di convergenza monotona in quanto ci sono casi in cui l'errore al posto di diminuire aumenta, come nel seguente caso:



Qui abbiamo che:

$$|e_1| \geq |e_0|$$

Nonostante questa cosa, le cose continuano a funzionare ugualmente

Scriviamo l'algoritmo:

Algoritmo del Metodo di Bisezione

Dati a, b, k_{max}, f e il "criterio di arresto"
Calcoliamo $f_a = f(a)$
Per $k = 1, 2, ..., k_{max}$
    $x = \frac{a+b}{2}$
    $f_x = f(x)$
    Criterio di arresto
    Se $sgn(f_a)f_x < 0$:
        $b = x$
    Altrimenti:
        $a = x$
        $f_a = f_x$

Facciamo delle osservazioni su quest'algoritmo.

Osservazione. Abbiamo messo $sgn(f_a)f_x$ al posto di $f_a f_x$ per evitare di dover trattare numeri troppo piccoli, in questo modo evitiamo complicanze dovute da ordini di grandezza e prodotti troppo piccoli.

Osservazione. Scrivere la media come $x = \frac{a+b}{2}$ può risultare pericoloso. Infatti è molto meglio scrivere la media come:

$$x = a + \frac{b-a}{2}$$

In questo modo è meno suscettibile al round-off.

Esempio 8 (Dove non funziona bene)**.** *Supponiamo di essere in aritmetica finita, in particolare in un'aritmetica in cui ci sono solamente tre cifre decimali e siano* $a = 0,982$ *e* $b = 0,984$*. Se volessimo andare a calcolare la media avremmo che:*

$$\frac{a+b}{2} = \frac{0,982 + 0,984}{2} = \frac{1,966}{2} = \frac{1,97}{2} = 0,985$$

Ma questa quantità appena ottenuta non solo non è la media, è addirittura sopra il più grande dei due. Con l'altro metodo:

$$a + \frac{b-a}{2} = 0,982 + \frac{0,984 - 0,982}{2} = 0,982 + \frac{0,002}{2} = 0,983$$

Questo è il valore che ci stavamo aspettando

Osservazione. C'è una sola valutazione di f per ogni iterazione

7.4 Criterio di Arresto

Oltre ad un primo criterio d'arresto, abbiamo possiamo definirne altri 2. La prima estremamente simile a quanto fatto nel primo modulo:

$$|f_x| < tol_f$$

Questa è una tolleranza sul "residuo"

L'altro criterio di arresto può essere definito come:

$$|b-a| < tol_r|a| + tol_a$$

Questa è una tolleranza sulla lunghezza dell'intervallo, in particolare tol_r prende il nome di **Tolleranza Relativa ad** a, mentre tol_a prende il nome di **Tolleranza Assoluta**. Vediamole singolarmente:

- Se $tol_a = 0$, allora abbiamo che il criterio di arresto può essere scritto come:

  $$\frac{|b-a|}{|a|} < tol_r$$

  Questo valore ci permette di controllare la vicinanza del punto al scelto ad a, ci permette quindi di liberare la distanza dal punto degli ordini di grandezza.

- Se invece poniamo $tol_r = 0$, abbiamo che il criterio di arresto può essere riscritto come:

  $$|b-a| < tol_a$$

  Questa invece è una tolleranza che mi controlla direttamente la dimensione dell'intervallo

7.5 Metodo di Newton

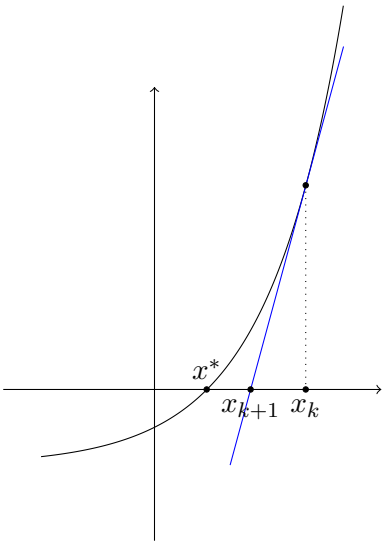
Sia $f : [a, b] \to \mathbb{R}$ e supponiamo x^* unico zero tale che $f(x^*)$ con $f'(x^*) \neq 0$ Partiamo da un $x_k[a, b]$ e cerchiamo x_{k+1}. Sviluppiamo in un intorno di x_k, allora abbiamo che:

$$f(x) = f(x_k) + f'(x_k)(x - x_k) + o(x_k)$$

Allora possiamo porre y come:

$$y = f(x_k) + f'(x_k)(x - x_k)$$

In questo modo poniamo:



In particolare, se prendiamo $x = x^*$, abbiamo che:

$$f(x^*) = f(x_k) + f'(x_k)(x^* - x_k) + o((x^* - x_k))$$

Non avendo tuttavia x^*, prendiamo come prossimo x_{k+1} l'ascissa sulla retta tangente corrispondente a $y = 0$, e così via. In questo modo abbiamo che:

$$0 = f(x_k) + f'(x_k)(x_{k+1-x_k}) \qquad \Rightarrow \qquad -\frac{f(x_k)}{f(x_{k+1})} = x_{k+1} - x_k$$

Da cui segue che:

$$x_{k+1} = x_k - \frac{f(x_k)}{f(x_{k+1})}$$

Questo algoritmo prende il nome di **Iterazione di Newton**. Scriviamone per bene l'algoritmo:

Algoritmo di Newton

Fissato $x_0 \in [a, n], k_{max}, f, df$ derivata di f
$f_0 = f(0)$
Per $k = 0, 1, ..., k_{max}$
    $df_k = df(x_k)$
    $x_{k+1} = x_k - \frac{f_k}{df_k}$
    Criterio di Arresto
        - Residuo: $f_{k+1} = f(x_{k+1}), |f_{k+1}| < tol$
        - $|x_{k+1} - x_k| \leq tol_r|x_k| + tol_a$

Generalmente abbiamo che questo metodo ha 2 possibili risultati, o converge velocemente, oppure

diverge sempre con la stessa velocità. Proprio per questo motivo non è strettamente necessario mettere un numero massimo di iterazioni abbastanza alto.

Il criterio di arresto segue lo stesso principio di quello precedente. Vediamo adesso perché funziona:

Esempio 9. *Supponiamo di avere* $x_{k+1} - x_k = 10^{-4}$. *Nel caso in cui abbiamo* $x_{k+1} = 2 \cdot 10^{-4}$ *e* $x_{k+1} = 10^{-4}$, *allora la prima cifra utile tra le due quantità è diversa, mentre se dovessi prendere* $x_{k+1} = 100,0002$ *e* $x_{k=100,0001}$, *allora la prima cifra che varierebbe è la sesta, ma la differenza resterebbe sempre la stessa. Nel secondo caso ci si può fermare, in quanto la variazione è minima rispetto alla differenza dei due. Nel primo caso no*

Osservazione. Se f è lineare, allora Newton converge dopo una sola iterazione

Esempio 10. *Sia* $f(x) = x^2 - 2$, *sappiamo allora che* $x = \sqrt{2}$. *Utilizzando il metodo di Newton è possibile calcolare* $\sqrt{2}$ *con:*

$$x_{k+1} = \frac{1}{x^k} + \frac{1}{2}x_k$$

Questa formula prende il nome di ***Formula di Erone***

Proposizione 7.5.1

Se $f \in X^2([a,b])$ e $\{x_k\}_{k\geq 0}$ di Newton converge, allora la convergenza è quadratica (cioè di ordine 2) per $f'(x^*) \neq 0$

Dimostrazione. Vogliamo mostrare che:

$$\lim_{k\to+\infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^2} = c$$

Per quanto fatto in precedenza, abbiamo che:

$$x_{k+1} - x^* = x_k - x^* - \frac{f(x_k)}{f'(x_k)} = \frac{1}{f'(x_k)}(f'(x_k)(x_k - x^*) - f(x_k) + f(x^*))$$

Ho aggiunto alla parentesi il termine $f(x^*)$ *in quanto* $f(x^*) = 0$. Otteniamo in questo modo che i termini dentro alla parentesi rappresentano uno sviluppo di Taylor, per cui:

$$f'(x_k)(x_k - x^*) - f(x_k) + f(x^*) = \frac{1}{2}f''(\xi_k)(x_k - x^*)^2 \qquad \xi_k \in (x_k, x^*)$$

In questo modo abbiamo ottenuto che:

$$x_{k+1} - x^* = \frac{1}{2}\frac{f''(\xi_k)}{f'(x_k)}(x_k - x^*)^2$$

Per cui abbiamo trovato che:

$$\frac{x_{k+1} - x^*}{(x_k - x^*)^2} = \frac{1}{2}\frac{f''(\xi_k)}{f'(x_k)} \xrightarrow[k\to+\infty]{\xi_k \to x^*} \frac{1}{2}\frac{f''(x^*)}{f'(x^*)} = c$$

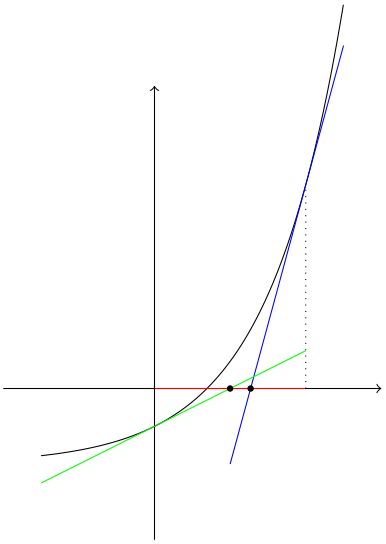
□

Osservazione. In realtà la convergenza è almeno quadratica, se infatti $f''(x^*) = 0$, si può avere convergenza più elevata, ma in generale la convergenza quadratica è più che sufficiente.

Proposizione 7.5.2: Semplificazione del Teorema di Newton-Kantovarich

Sia $f \in C^2([a,b])$ con f convessa, o concava in $[a,b]$. Supponiamo $f(a)f(b) < 0$ e che le tangenti di a e di b si intersechino l'asse delle ascisse in un punti interni ad $[a,b]$, allora l'iterazione di Newton converge globalmente in $[a,b]$, cioè $\forall x \in [a,b]$, nell'unico zero $x^* \in [a,b]$

Considerazioni Aggiuntive. *L'unicità dello zero segue dal fatto che f sia convessa o concava in tutto l'intervallo. Graficamente abbiamo che:*

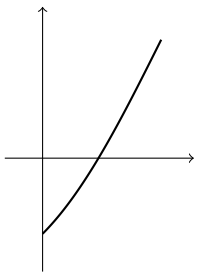


Esercizio. Dimostrare che il metodo di Newton applicato a $f(x) = x - \cos x$ converge globalmente in $[0, \frac{\pi}{2}]$

Soluzione. Vogliamo studiare la convergenza ad x^* tale che $f(x^*) = 0$. Sappiamo che $f(x) = x - \cos x \in C^\infty$. Per cui abbiamo che:

$$f'(x) = 1 + \sin x > 0 \qquad \text{e} \qquad f''(x) = \cos x \geq 0$$

Cioè abbiamo che la derivata prima è crescente, mentre la seconda è positiva e continua, quindi la funzione è convessa nel dominio.



Sappiamo inoltre che:

$$f(a) = f(0) = 0 - \cos 0 = -1 < 0 \qquad \text{e} \qquad f(b) = f\left(\frac{\pi}{2}\right) = \frac{\pi}{2} - \cos\frac{\pi}{2} = \frac{\pi}{2} > 0$$

Viene quindi soddisfatta l'ipotesi che $f(a)f(b) < 0$. Vediamo allora se le tangenti toccano l'asse delle ascisse in $[a, b]$:

$$x_0 = 0 \quad x_1 = \frac{f(x_0)}{x'(x_0)} = 0 - \frac{-1}{1} = 1 \in \left[0, \frac{2}{\pi}\right]$$
$$x_0 = \frac{\pi}{2} \quad x_1 = \frac{f(\frac{\pi}{2})}{f'(\frac{\pi}{2})} = \frac{\pi}{2} - \frac{\frac{\pi}{2}}{2} = \frac{\pi}{4} \in \left[0, \frac{\pi}{2}\right]$$

Per la proposizione precedente si ha convergenza globale in $[0, \frac{\pi}{2}]$ ■

7.6 Analisi di Convergenza di Newton

Definizione 7.6.1: Funzione Lipschitziana

Sia $g : X \to \mathbb{R}$. g si dice **Lipschitziana** con costante $L < 0$ in X, e si indica con $g \in Lip_L([a, b])$ se:

$$\forall x, y \in X : |g(x) - g(y)| \leq L(x - y)$$

Lemma 7.6.2

Sia $f : [a, b] \to \mathbb{R}$ con $f' \in Lip_L([a, b])$, allora $\forall x, y \in [a, b]$ vale:

$$|f(y) - f(x) - f'(x)(y - x)| \leq \frac{L}{2}|x - y|^2$$

Dimostrazione. Partiamo dalla prima parte della disuguaglianza, senza valore assoluto:

$$f(y) - f(x) - f'(x)(y - x) = \int_x^y f'(t)dt - \int_x^y f'(x)dt = \int_x^y f'(t) - f'(x)dt$$

Facciamo un cambiamento di variabile:

$$t = x + \tau(y - x) \quad \text{per } \tau \in [0, 1] \quad \Rightarrow \quad dt = (y - x)d\tau$$

In questo modo abbiamo che:

$$f(y) - f(x) - f'(x)(y - x) = \int_x^y f'(t) - f'(x)dt = \int_0^1 (f'(x + \tau(y - x)) - f'(x))(y - x)d\tau$$

Quindi, con i valori assoluti, abbiamo che:

$$|f(y) - f(x) - f'(x)(y - x) \leq |y - x| \int_0^1 |f'(x + \tau(y - x) - f('x))|d\tau$$

Su quest'ultima parte dentro l'integrale può essere applicata la definizione di funzione lipschitziana con opportuno L, per cui:

$$|f'(x + \tau(y - x) - f('x))| \leq L|x + \tau(y - x) - x| = L|\tau(y - x)|$$

Per cui abbiamo che

$$|f(y) - f(x) - f'(x)(y - x) \leq |y - x| \int_0^1 |f'(x + \tau(y - x) - f('x))|d\tau \leq |y - x|^2 L \int_0^1 \tau d\tau = \frac{1}{2}L(y - x)^2$$

□

Teorema 7.6.3: Convergenza dell'Algoritmo di Newton

Sia $f : [a, b] \to \mathbb{R}$ con $f \in Lip_L([a, b])$. Supponiamo che esista $\rho > 0$ tale che:

$$|f'(x)| \geq \rho, \forall x \in [a, b]$$

Supponiamo inoltre che $\exists x^* \in (a, b)$ tale che $f(x^*) = 0$ radice semplice. Allora esiste $\eta > 0$ tale che:

1. Se $|x_0 - x^*| < \eta$, allora l'iterazione di Newton è ben definita e converge a x^*
2. Vale $|x_{k+1} - x^*| \leq \frac{L}{2\rho}|x_k - x^*|^2$

Considerazioni Aggiuntive. *Questo è un risultato importante in quanto ci dice che:*

- *Possiamo usare il fatto che la funzione sia Lipschitziana per avere una sufficiente regolarità di f*
- *La condizione su ρ ci dice che la derivata prima non deve essere troppo vicino a 0, deve essere sufficiente lontana. Questa condizione è fondamentale in quanto previene la convergenza ad un punto $x \neq x^*$ che assume valori molto vicini allo zero*
- *x^* deve essere una radice semplice della funzione*
- *La relazione di 2. vale indipendentemente dalla convergenza del metodo, quindi continua a valere sia se c'è convergenza, sia se c'è divergenza.*
- *La quantità $\frac{L}{2\rho}$ ci dice come può essere maggiorata la distanza di x^* da x_{k+1}. Questa quantità dipende da L al numeratore, cioè dipende proporzionalmente da quanto $f'(x)$ cresce, e da ρ al denominatore.*
- *Notiamo che l'errore che passa al passo $k+1$ è maggiorato da quella costante per l'errore del passo precedente al quadrato. Quindi c'è una relazione quadratica tra il passo k e il passo $k+1$. Quindi, se ho convergenza, ho una convergenza quadratica (l'abbiamo già visto prima). Se non ho convergenza, allora ho divergenza che procede alla stessa velocità*

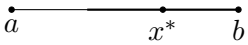
Dimostrazione. [2] Iniziamo con il dimostrare la disuguaglianza. Sfruttando il lemma precedente otteniamo che:

$$\begin{aligned} |x_{k+1} - x^*| &= \left|(x_k - x^*) - \frac{f(x_k)}{f'(x_k)}\right| = \frac{1}{|f'(x_k)|}|f'(x_k)(x_k - x^*) - f(x_k) + f(x^*)| \\ &= \frac{1}{|f'(x_k)|^2}\frac{1}{2}|x_k - x^*|^2 < \frac{1}{\rho}\frac{L}{2}|x_k - x^*|^2 \end{aligned}$$

[1] Vogliamo mostrare che preso $\tau \in (0, 1)$ vale che:

$$|x_{k+1} - x^*| \leq \tau|x_n - x^*| < \eta$$

dove poniamo $\eta = \min\{\frac{2\rho}{L}\tau, \hat{\eta}\}$, con $\hat{\eta}$ il più grande raggio del disco centrato in x^* e tutto in $[a, b]$



Dimostriamolo per induzione. Se $|x_0 - x^*| < \eta$ allora abbiamo che, per $k = 1$:

$$|x_1 - x^*| \leq \frac{L}{2\rho}|x_0 - x^*|^2 = \frac{L}{2\rho}|x_0 - x^*| \cdot |x_0 - x^*|$$

Dalla definizione di η, abbiamo che $|x_0 - x^*| < \eta \leq \frac{2\rho}{L}\tau$, per cui otteniamo che:

$$|x_1 - x^*| \leq \frac{L}{2\rho}|x_0 - x^*| \cdot |x_0 - x^*| \leq \frac{L}{2\rho}\frac{2\rho}{L}\tau|x_0 - x^*| < |x_0 - x^*| < \eta$$

Supponiamo adesso $|x_k - x^*| < \eta$ e facciamo vedere che vale anche per x_{k+1}:

$$\begin{aligned}|x_{k+1} - x^*| &\leq \frac{L}{2\rho}|x_k - x^*|^2 = \frac{L}{2\rho}|x_k - x^*| \cdot |x_k - x^*| \leq \frac{L}{2\rho}\frac{2\rho}{L}\tau|x_k - x^*| = \\ &= \tau|x_k - x^*| < |x_k - x^*| < \eta\end{aligned}$$

□

7.7 Esercizi su Newton

Esercizio (Esame 8/01/2021)**.** Sia $f(x) = e^{-x} - \sin x$. Mostrare che l'iterazione di Newton converge a $x^* \approx 0,5885$ supponendo di aver scelto x_0 in modo opportuno

Osservazione. Il fatto che sappiamo $x^* = 0,5885$ ci da soltanto un'indicazione di dove sta la soluzione

Soluzione. Basta far vedere che valgono le ipotesi del teorema della convergenza locale. Calcoliamo la derivata prima:

$$f'(x) = -e^{-x} - \cos x$$

Se prendiamo $x \in [0, \frac{\pi}{2}]$, abbiamo che:

$$|f'(x)| = |e^{-x} + \cos x| = e^{-x} + \cos x \geq e^{-x} \geq e^{-\pi/2} = \rho$$

Inoltre sappiamo che $f \in C^\infty([0, \frac{\pi}{2}])$, quindi è sicuramente di classe C^2. Inoltre la sua derivata seconda è:

$$f''(x) = e^{-x} + \sin(x)$$

È limitata in $[0, \frac{\pi}{2}]$. Quindi per il teorema di convergenza locale, locale converge localmente (per un x_0 scelto in maniera opportuna) ■

Esercizio (Esame 31/01/2022)**.** Data $f(x) = \sin(x) - \frac{\pi}{2}$ con $x \in [\frac{1}{3}\pi, \frac{3}{4}\pi]$ mostrare che il metodo di Newton converge localmente

Soluzione. Si segue esattamente lo stesso procedimento dell'esercizio precedente ■

Esercizio (Esame 13/07/2020)**.** Sia $f \in C^2([a,b])$ con x^* unico zero di f in $[a,b]$. Supponiamo che il metodo di Newton generi $\{x_k\}_{k\geq 0}$ successione ben definita in $[a,b]$ e sia $M > 0$ tale che:

$$\frac{1}{2}\left|\frac{f''(y)}{f'(x)}\right| \leq M \qquad \forall y, x \in [a,b]$$

1. Sfruttando Taylor intorno a x^* mostrare che vale:

$$|x^* - x_{k+1}| \leq M|x^* - x_k|^2$$

2. Determinare un intorno $I = I(M)$ di x^* tale che si abbia convergenza per $x_0 \in I(M)$

Soluzione. 1 Questo punto prevede sostanzialmente di rifare quanto fatto nella dimostrazione del teorema:

$$\begin{aligned}|x_{k+1} - x^*| &= \left|x_k - x^* - \frac{f(x_k)}{f'(x_k)}\right| = \frac{1}{|f'(x_k)|}|f'(x_k)(x_k - x^*) - f(x_k) - f(x^*)| \\ &= \frac{1}{2}\left|\frac{f''(\xi_x)}{f'(x_k)}\right|(x_k - x^*)^2 \qquad \text{per } \xi_x \in (x_k, x^*)\end{aligned}$$

Tuttavia, ricordando che $\xi_x, x_k \in [a,b]$, valgono le ipotesi date, quindi:

$$|x_{k+1} - x^*| = \frac{1}{2}\left|\frac{f''(\xi_x)}{f'(x_k)}\right|(x_k - x^*)^2 \leq M(x_k - x^*)^2$$

2 Fissiamo $\tau \in [0,1]$ e prendiamo η come:

$$\eta = \min\left\{\frac{1}{M}\tau, \hat{\eta}\right\}$$

Dove $\hat{\eta}$ è il raggio del più grande disco centrato in x^* e contenuto in $[a,b]$. Se prendo x_0 in modo che:

$$x_0 - x^* < \eta \leq \frac{1}{M}\tau$$

Allora possiamo procedere come nel teorema:

$$\begin{aligned}|x_1 - x^*| \leq M|x_0 - x^*|^2 = M|x_0 - x^*| \cdot |x_0 - x^*| < M\eta \cdot |x_0 - x^*| \leq \\ \leq M\frac{1}{M}\tau|x_0 - x^*| = \tau|x_0 - x^*| < \eta \leq \frac{1}{M}\tau\end{aligned}$$

Procediamo allora per induzione per x_{k+1}, assumendo vero per x_k:

$$|x_{k+1} - x^*| \leq M|x_k - x^*|^2 \leq \tau|x_k - x^*| < |x_k - x^*| < \eta$$

Da cui segue che l'intervallo voluto è:

$$I = (x^* - \eta, x^* + \eta)$$

■

Osservazione. Nel caso in cui x^* abbia molteplicità $m > 1$, se il metodo di Newton converge, la sua convergenza è lineare:

$$|x_{k+1} - x^*| \leq C|x_k - x^*| \qquad \text{con } C = 1 - \frac{1}{M}$$

7.8 Varianti (Metodi Quasi Newton)

Con Newton abbiamo che possiamo approssimare x_{k+1} come:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Tuttavia non sempre abbiamo la derivata prima, quindi dobbiamo approssimarla:

$$x_{k+1} = x_k - \frac{f(x_k)}{q_k}$$

Ci sono molti modi per approssimare la derivata prima:

- **Newton Inesatto**: Questo metodo consiste nel sostituire la derivata prima con:

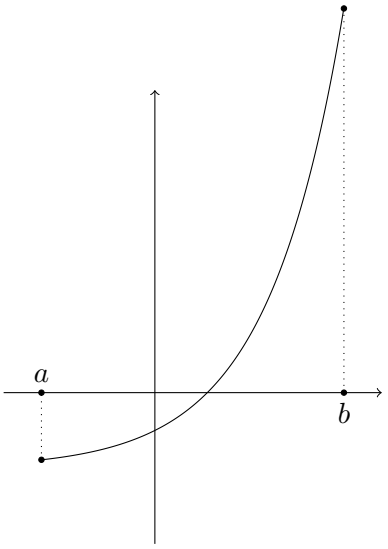
  $$q_k = \frac{f(x_k + h) - f(x_k)}{h}$$

  Questo metodo quindi prevede una valutazione di f due volte per iterazione. Quindi possiamo dire che l'accuratezza del metodo è dell'ordine di $\mathcal{O}(h)$ e dipende dalla scelta di q_k

- **Metodo delle Corde**: qui il valore q_k viene calcolato una volta soltanto, ponendo:

  $$q_k = q = \frac{f(b) - f(a)}{b - a}$$

  Il valore q corrisponde al coefficiente angolare della retta che unisce (a) e $f(b)$:

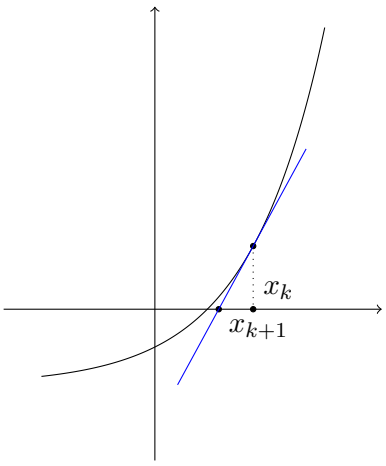
  

  Se siamo fortunati abbiamo anche la pendenza giusta. Volendo si può dimostrare che la pendenza è lineare, cioè perdiamo totalmente la convergenza quadratica di Newton.

- **Metodo delle Secanti**: Prendiamo q_k come:

  $$q_k = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

  In questo modo dobbiamo memorizzare i valori vecchi della funzione, così da dover valutare la funzione una volta soltanto per iterazione:



Ad ogni iterazione uso sia $f(x_k)$ sia $f(x_{k-1})$. Quindi l'algoritmo sarà:

Algoritmo del Metodo delle Secanti

Dati x_0, x_1

Per $k = 0, 1, ...$

$$x_{k+1} = x_k - \frac{f(x_k)}{\left(\frac{f(x_k)-f(x_{k-1})}{x_k-x_{k-1}}\right)} = x_k - \frac{f_{new}}{\left(\frac{f_{new}-f_{old}}{x_k-x_{k-1}}\right)}$$

$$f_{new} = f(x_{k+1})$$

$$f_{old} = f(x_k)$$

Nel metodo delle secanti ho bisogno di 2 dati iniziali, possibilmente tali che $f(x_0)f(x_{-1}) < 0$

Osservazione. In Matlab, la funzione `fzero` usa un ibrido del metodo di bissezione (che serve per avvicinarsi allo zero della funzione) e del metodo delle secanti (per arrivare nell'effettivo alla convergenza)

8 Zeri di Polinomi

8.1 Problema e Prime Soluzioni

Il problema consiste nel trovare un'approssimazione di x^* in modo tale che $p_n(x^*) = 0$, dove:

$$p_n(x^*) = a_0 + a_1 x + \cdots + a_n x^n \qquad (a_n \neq 0)$$

Questo è un problema estremamente sensibile in quanto non è detto che ci siano radici reali, quindi il problema potrebbe non avere senso.

Fissiamo un x_0 e scriviamo la ricorrenza di Newton:

$$x_{k+1} = x_k - \frac{p_n(x_k)}{p_n'(x_k)}$$

Restano i problemi della valutazione di p_n e p_n' in x_k.
Per ottenere $p_n(x_k)$, ci sono diversi modi per ottenerli:

- Possiamo pensare di fare i conti esplicitamente, però bisogna fare potenze su potenze, quindi è meglio evitare.

- Possiamo salvarci volta per volta il valore di x_k^j:

  Algoritmo per la Valutazione di un Polinomio

  Fisso $s = 1$
  $p = a_0$
  Per $j = 1, ..., n$
  $\quad s = s \cdot x_k$
  $\quad p = p + a_j s$

  In questo modo, alla fine del ciclo, abbiamo che $p = p_n(x_k)$. In questo modo, rispetto a prima, sfruttiamo i conti già fatti e, avendo per ogni iterazione 3 operazioni, quest'algoritmo ha un costo computazionale di $3n$ flops.

- Possiamo utilizzare la Regola di Horner, secondo cui possiamo scrivere $p_n(x)$ come:

  $$p_n(x) = a_0 + x(a_1 + x(a_2 + \cdots + x(a_{n-1} + a_n x) \cdots))$$

  In questo modo l'algoritmo diventa:

  Algoritmo per la Valutazione di un Polinomio con la regola di Horner

  $b_0 = a_0$
  Per $j = 1, ..., n$
  $\quad b_j = b_{j-1} x_k + a_{n-j}$

  In questo modo abbiamo che $b_n = p_n(x_k)$. Notiamo che questo corrisponde a risolvere un

sistema lineare con una matrice bidiagonale:

$$\begin{pmatrix} 1 & -x_k & & & & & \\ & 1 & -x_k & & & & \\ & & 1 & -x_k & & & \\ & & & \ddots & \ddots & & \\ & & & & 1 & -x_k & \\ & & & & & 1 & -x_k \\ & & & & & & 1 \end{pmatrix} \begin{pmatrix} b_n \\ b_{n-1} \\ b_{n-2} \\ \vdots \\ b_2 \\ b_1 \\ b_0 \end{pmatrix} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-2} \\ a_{n-1} \\ a_n \end{pmatrix}$$

In totale ha un costo computazionale pari a $2n$. Questo procedimento è lo stesso che fa Matlab.

Cerchiamo ora di capire come possiamo fare per ottenere $p_n'(x_k)$. Possiamo scrivere:

$$p_n(x) = (x - x_n)\hat{p}_{n-1}(x) + b_n$$

Dove abbiamo che:

$$\hat{p}_{n-1}(x) = b_0 x^{n-1} + b_1 x^{n-2} + \cdots + b_{n-1}$$

Volendo è possibile verificare che vale l'uguaglianza termine per termine, infatti (per $k = n-1$ per esempio):

$$a_{n-1}x^{n-1} = x(b_1 x^{n-2} + (-x_n)b_0 x^{n-1}) \qquad \Leftrightarrow \qquad a_{n-1} = b_1 - x_n b_0$$

Questo coincide esattamente con l'iterazione $j = 1$. Per cui abbiamo che vale:

$$p_n'(x) = \hat{p}_{n-1}(x) + (x - x_k)\hat{p}_{n-1}(x) \qquad \Rightarrow \qquad p_n'(x)\big|_{x=x_k} = \hat{p}_{n-1}(x)\big|_{x=x_k}$$

Quindi possiamo usare nuovamente la regola di Horner:

Algoritmo per la Derivata in un Punto di un Polinomio

Poniamo $c_0 = b_0$
Per $j = 1, ..., n-1$
$\quad c_j = c_{j-1}x_k + b_j$

In questo modo otteniamo che $c_{n-1} = \hat{p}_{n-1}(x_k) = p_n'(x_k)$. Possiamo allora scrivere un unico algoritmo per calcolare sia il valore della funzione, sia quello della derivata prima:

Algoritmo per la Valutazione della Funzione e della Derivata in un Punto di un Polinomio

Poniamo $b_0 = a_n$ e $c_0 = b_0$
Per $j = 1, ..., n-1$
$\quad b_j = b_{j-1}x_k + a_{n-k}$
$\quad c_j = c_{j-1}x_k + b_j$
$b_n = b_{n-1}x_k + a_0$

Otteniamo quindi i valori di p_n e di p_n' nella iterata corrente x_k di Newton.

Vediamo dei primi risultati:

Teorema 8.1.1

Sia p_n un polinomio a coefficienti reali con radici $\xi_1 \geq \xi_2 \geq \cdots \geq \xi_n$ tutte reali, allora il metodo di Newton genera una successione $\{x_k\}_{k\geq 0}$ convergente a ξ_1 in modo monotono strettamente decrescente per ogni $x_0 > \xi_1$

Osservazione. Con questo teorema, quello che abbiamo sostanzialmente fatto è stato spostare il problema della convergenza a quello della scelta di x_0

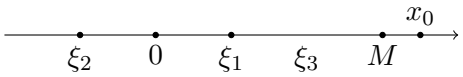
Teorema 8.1.2: Localizzazione delle Radici

Le radici ξ_i con $i \in \{1, ..., n\}$ di un polinomio $p_n(x) = a_0 + a_1 x + \cdots + a_n x^n$ con $a_n \neq 0$ soddisfano:

$$|\xi_i| \leq \max\left\{1, \sum_{j=0}^{n-1} \frac{|a_j|}{|a_n|}\right\} \equiv M_1$$

$$|\xi_i| \leq \max\left\{\frac{|a_0|}{|a_n|}, 1 + \frac{|a_1|}{|a_n|}, \cdots, 1 + \frac{|a_{n-1}|}{|a_n|}\right\} \equiv M_2$$

Considerazioni Aggiuntive. *Tutti questi valori noi li abbiamo, in quanto corrispondono ai coefficienti dei polinomi. Quindi sappiamo trovare tutti i valori e di conseguenza sappiamo dove porre x_0, infatti, posto $M = \max\{M_1, M_2\}$ ci basta porre $x_0 > M$:*



Osservazione. Supponiamo ξ_1 radice reale ottenuta di $p_n(x)$, polinomio avente coefficienti e radici reali $\xi_1 > \xi_i, \forall i \in \{2, ..., n\}$. Supponiamo di voler trovare le altre radici di $p_n(x)$. Allora possiamo definire un polinomio $q_{n-q}(x)$ come:

$$q_{n-1} = \frac{p_n(x)}{x - \xi_1}$$

Numericamente, quello che stiamo facendo è sostanzialmente è ottenere $\tilde{\xi}_1$, approssimazione di ξ_1. In particolare abbiamo che:

$$q_{n-1}(x) = \frac{p_n(x)}{x - \tilde{\xi}_1} = a_n(x - \xi_n)(x - \xi_{n-1}) \cdots \frac{x - \xi_1}{x - \tilde{\xi}_1}$$

è una funzione razionale. Quindi, numericamente parlando, questa è una procedura pericolosa. Questo procedimento prende il nome di **Deflazione**

Prima di dare la dimostrazione del teorema 8.1.2 diamo questa definizione

Definizione 8.1.3: Matrice Compagna

Sia $p_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0$ con $a_n \neq 0$. Si definisce la **Matrice Compagna** o **Matrice Companion** la matrice $n \times n$:

$$C = \begin{pmatrix} 0 & & & & -\frac{a_0}{a_n} \\ 1 & 0 & & & -\frac{a_1}{a_n} \\ & 1 & \ddots & & \vdots \\ & & \ddots & 0 & \vdots \\ & & & 1 & -\frac{a_{n-1}}{a_n} \end{pmatrix}$$

Il suo polinomio caratteristico è $\varphi(\lambda) = \det(C - \lambda I)$

Proposizione 8.1.4

Vale l'uguaglianza:

$$\varphi(\lambda) = \frac{(-1)^n}{a_n} p_n(\lambda)$$

Cioè le radici di p_n corrispondono con gli autovalori di C.

Diamo adesso la dimostrazione del teorema 8.1.2:

Dimostrazione. Il teorema è una applicazione di Gerschgorin alla matrice C:

$$\mathcal{G}_1^{(r)} : |\lambda| \leq \frac{|a_0|}{a_n} \qquad \text{e} \qquad \mathcal{G}_i^{(r)} : |\lambda| \leq \frac{|a_{i-1}|}{|a_n|} + 1 \quad \forall i \in \{2, ..., n\}$$

Per quanto riguarda $i = n$ abbiamo che:

$$\mathcal{G}_n^{(r)} : \left|\lambda + \frac{a_{n-1}an}{\leq} 1\right|$$

Ma abbiamo che:

$$\left|\lambda + \frac{a_{n-1}}{a_n}\right| \geq |\lambda| - \left|\frac{a_{n-1}}{a_n}\right| \qquad \Rightarrow \qquad |\lambda| \leq 1 + \frac{|a_{n-1}|}{|a_n|}$$

Quindi è verificata la seconda disuguaglianza del teorema.

Sappiamo inoltre che:

$$\mathcal{G}_i^{(r)} : |\lambda| \leq 1 \qquad \forall i \in \{1, ..., n-1\}$$

Per quanto riguarda $i = n$, possiamo dire che:

$$\mathcal{G}_n^{c} : \left|\lambda + \frac{|a_{n-1}|}{|a_n|}\right| \leq \sum_{k=0}^{n-2} \frac{|a_k|}{|a_n|}$$

Sapendo poi che:

$$\left|\lambda + \frac{a_{n-1}}{a_n}\right| \geq |\lambda| - \left|\frac{a_{n-1}}{a_n}\right| \qquad \Rightarrow \qquad |\lambda| \leq \sum_{k=0}^{n-1} \frac{|a_k|}{a_n}$$

Quindi è verificata anche la prima disuguaglianza del teorema. □

8.2 Questioni di Stabilità

Andiamo a studiare come variano le radici dei polinomi quando i coefficienti dei polinomi vengono perturbati. Consideriamo $p(x)$ con radice semplice ξ. Vogliamo studiare il comportamento di $\xi(\varepsilon)$, radice perturbata, tale che:

$$\xi(\varepsilon) \xrightarrow{\varepsilon \to 0} \xi = \xi(0)$$

Definiamo allora p_ε come:

$$p_\varepsilon(x) = p(x) + \varepsilon g(x)$$

Dove $g(x)$ è un polinomio di grado al più n, per esempio $g(x) = \gamma_2 x^2$. Sia $\xi(\varepsilon)$ radice di p_ε, cioè:

$$p_\varepsilon(\xi(\varepsilon)) = 0$$

Deriviamo adesso rispetto a ε:

$$0 = \frac{d}{d\varepsilon} p_\varepsilon(\xi(\varepsilon)) = \frac{d}{d\varepsilon} p(\xi(\varepsilon)) + g(\xi(\varepsilon)) + \varepsilon \frac{d}{d\varepsilon} g(\xi(\varepsilon)) = p'(\xi(\varepsilon))\xi'(\varepsilon) + g(\xi(\varepsilon)) + \varepsilon \frac{d}{d\varepsilon} g(\xi(\varepsilon)) = 0$$

Da cui, sostituendo per $\varepsilon = 0$, si ottiene che:

$$p'(\xi(0))\xi'(0) + g(\xi(0)) = 0$$

Da cui si ottiene che:

$$\xi'(0) = -\frac{g(\xi(0))}{p'(\xi(0))}$$

Dove $\xi(0)$ è la radice non perturbata di $p(x)$. Ma allora abbiamo che:

$$\xi(\varepsilon) \xrightarrow{\varepsilon \to 0} \qquad \Rightarrow \qquad \xi(\varepsilon) = \xi(0) + \xi'(0)\varepsilon + o(\varepsilon) \quad \text{per } \varepsilon \to 0$$

In questo modo otteniamo che:

$$\xi(\varepsilon) - \xi(0) = \xi'(0)\varepsilon + o(\varepsilon) + o(\varepsilon) \approx -\frac{g(\xi(0))}{p'(\xi(0))}$$

Esempio 11 (Wilkinson)**.** *Consideriamo il polinomio* $p(x) = (x-1)(x-2)\cdots(x-20)$ *di grado* 20*. Consideriamo* $\xi = 20$*, allora abbiamo che:*

$$p'(20) = 19! \qquad e \qquad a_{19} = -210$$

Consideriamo quindi $g(x) = a_{19}x^{19}$*, allora il polinomio* $p_\varepsilon(x) = p(x) + \varepsilon g(x)$ *ha come coefficiente la quantità* $a_{19} + \varepsilon a_{19}$*. Da quanto appena fatto abbiamo che:*

$$\xi(\varepsilon) - \xi(0) \approx -\frac{-210 \cdot 20^{19}}{19!}\varepsilon \approx 10^{10}\varepsilon$$

Se prendiamo $\varepsilon = 10^{-10}$ *allora abbiamo che:*

$$\xi(\varepsilon) - \xi(0) \approx 1$$

Quindi la radice si è completamente spostata, anche avendo perturbato di poco

9 Iterazione di Punto Fisso

9.1 Presentazione del problema

Sia $f : [a, b] \to \mathbb{R}$ tale che:

$$f(x) = 0 \qquad \Leftrightarrow \qquad \Phi(x) = x$$

Per qualche Φ. In particolare, Φ è scelta in modo che per $f(x^*) = 0$ si abbia $x^* = \Phi(x^*)$. Dalla scrittura $\Phi(x^*) = x^*$ si ottiene l'iterazione del tipo $x_{k+1} = \Phi(x_k)$ per un qualche x_0

Osservazione. La funzione Φ non è numericamente determinata

Teorema 9.1.1: Convergenza

Sia x_0 assegnato e sia l'iterazione $x_{k+1} = \Phi(x_k)$ tale che:

1. $\Phi : [a, b] \to [a, b]$ e che $\Phi \in C^1([a, b])$
2. Esiste $K < 1$ tale che $|\Phi'(x)| \leq K$, $\forall x \in [a, b]$, cioè Φ è una contrazione

Allora Φ ha un unico punto fisso x^* in $[a, b]$ e l'iterazione $\{a_k\}_{k\in\mathbb{N}}$ converge in x^* $\forall x_0 \in [a, b]$ (cioè si ha convergenza globale). Inoltre:

$$\lim_{x\to+\infty} \frac{x_{k+1} - x^*}{x_k - x^*} = \Phi(x^*)$$

Cioè lo stesso $\Phi(x^*)$ rappresenta il fattore asintotico di convergenza

Osservazione. Il metodo è fortemente consistente

Dimostrazione. Le ipotesi del punto 1 ci assicurano l'esistenza del punto fisso x^*. Infatti, definiamo allora F come:

$$F(x) = x - \Phi(x)$$

Notiamo allora che:

$$F(a) = a - \Phi(a) \leq 0 \qquad \text{e} \qquad F(b) = b - \Phi(b) \geq 0$$

Sappiamo che F è una funzione continua, quindi $\exists x^* \in [a, b] : F(x^*) = 0$.

Dimostriamo ora l'unicità. Per assurdo supponiamo esistano α_1, α_2 distinti punti fissi, allora abbiamo che:

$$|\alpha_1 - \alpha_2| = |\Phi(\alpha_1) - \Phi(\alpha_2)|$$

Sfruttiamo il fatto che Φ sia di classe C^1, allora esiste $\xi \in [\alpha_1, \alpha_2]$ tale che:

$$|\alpha_1 - \alpha_2| = |\Phi(\alpha_1) - \Phi(\alpha_2)| = |\Phi'(\xi)||\alpha_1 - \alpha_2| \leq K|\alpha_1 - \alpha_2| < |\alpha_1 - \alpha_2|$$

Per la convergenza usiamo Taylor:

$$x_{k+1} - x^* = \Phi(x_k) - \Phi(x^*) = \Phi'(\xi_k)(x_k - x^*) \qquad \text{per } \xi \in (x_k, x^*)$$

In questo modo abbiamo che:

$$|x_k - x^*| \leq K|x_k - x^*| \leq K^{k+1}|x_0 - x^*| \xrightarrow{k\to+\infty} 0$$

Questo è vero in quanto abbiamo che $|K| < 1$. Infine, dall'espressione precedente abbiamo che:

$$\frac{|x_{k+1} - x^*|}{|x_k - x^*|} = |\Phi'(\xi_k)| \xrightarrow[\Phi \text{ continua}]{k \to +\infty} |\Phi'(x^*)|$$

Dove abbiamo che $\xi_k \to x^*$ □

Teorema 9.1.2: di Ostrowski

Sia x^* punto fisso di Φ continua e derivabile in un intorno di I di x^*. Se $|\Phi'(x^*)| < 1$, allora $\exists \delta \in \mathbb{R}^+$ tale che l'iterazione $(x_k)_{k \in \mathbb{N}}$ converge a x^*, $\forall x \in]x^* - \delta, x^* + \delta[$

Osservazione. La condizione definita è sufficiente, ma è anche necessaria. Infatti, se $|\Phi(x)| > 1$ si può dimostrare che l'iterazione diverge. Si ha inoltre convergenza locale

Esercizio. Sia $f(x) = e^{-x} - x^2$ con $x^* \approx 0,7$. Studiare la convergenza dell'iterazione di punto fisso con $\Phi(x) = e^{-x/2}$. Quindi $x_{k+1} = e^{-\frac{xk}{2}}$

Soluzione. Andiamo a calcolarne la derivata:

$$\Phi'(x) = -\frac{1}{2} e^{-x/2}$$

Allora, per $x \in [0, 1]$, abbiamo che:

$$|\Phi'(x)| = \frac{1}{2} e^{-x/2} \leq \frac{1}{2} < 1$$

Possiamo dire che $\Phi([0, 1]) \subseteq [0, 1]$?. Sappiamo che $\Phi \in C^2$, quindi:

$$0 \leq \Phi(x) = e^{-x/2} \leq 1 \qquad textperx \in [0, 1]$$

Quindi per il teorema di convergenza visto, converge $\forall x \in [0, 1]$ ■