CALCOLO NUMERICO

WIKIBOOKS
Libri liberi per un mondo aperto

Calcolo numerico

it.wikibooks.org
2025

Questo testo proviene dal sito
https://it.wikibooks.org/wiki/Calcolo_numerico

La versione originale del testo si trovava su WikiToLearn. Una copia è disponibile su
https://web.archive.org/web/20200919224247/https:/it.wikitolearn.org/Corso:Calcolo_Numerico_I1

Autori principali:
Michela Montrasio, utenti di WikiToLearn

Questo libro è aggiornato al
13 aprile 2025

In copertina:
Computer al Computer History Museum. *Autore:* Marcin Wichary; *licenza:* CC BY 2.0; *fonte:*
https://commons.wikimedia.org/wiki/File:CHM_Artifacts_(3300652765).jpg

Wikibooks non dà garanzie sulla validità dei suoi contenuti. Per i dettagli vedi:
https://it.wikibooks.org/wiki/Wikibooks:General_disclaimer

Quest'opera è distribuita con licenza **Creative Commons Attribuzione - Condividi allo stesso modo 4.0 Internazionale**.
Per leggere una copia della licenza visita il sito:
https://creativecommons.org/licenses/by-sa/4.0/deed.it

Indice

1 Aritmetica floating point **1**
1.1 Introduzione all'aritmetica floating point 1
1.2 Cos'è l'aritmetica floating point 4
1.3 Rappresentazione floating dei numeri 8
1.4 Operazioni aritmetiche con il calcolatore 11
1.5 Teoria dell'errore nella sua generalità 16
1.6 Condizionamento e stabilità 18
1.7 Comportamento dell'errore algoritmico e dell'errore analitico 28

2 Sistemi lineari **31**
2.1 Norme di vettori e di matrici 31
2.2 Sistemi lineari e errore inerente (condizionamento) 39
2.3 Sistemi lineari su un calcolatore 44
2.4 Fattorizzazione LU (eliminazione gaussiana) 45
2.5 Tecniche del pivot 49
2.6 Fattorizzazione di Cholesky 55
2.7 Fattorizzazione QR 61
2.8 Metodi iterativi 69
2.9 Principali metodi iterativi 75
2.10 Metodi di rilassamento 81

3 Localizzazione degli autovalori **85**
3.1 Localizzazione degli autovalori 85
3.2 Calcolo di autovalori estremi 87
3.3 Metodo della potenza 88
3.4 Metodo della potenza inversa 91
3.5 Casi particolari 92

4 Radici di equazioni non lineari **97**
4.1 Osservazioni introduttive 97
4.2 Bisezione corde secanti Newton 98
4.3 Metodo di Brent 102
4.4 Metodi del punto fisso 105
4.5 Condizionamento test e stabilità dei metodi di punto fisso 109
4.6 Radici di sistemi non lineari 111

5 Interpolazione polinomiale **113**
5.1 Osservazioni introduttive 113
5.2 Altre scritture dei polinomi interpolanti 116
5.3 Stabilità condizionamento e convergenza 121
5.4 Interpolazione polinomiale a tratti (spline) 125
5.5 Approssimazione media dei minimi quadrati 131
5.6 Approssimazione dei minimi quadrati con funzioni continue 137

6 Formule di quadratura **141**
6.1 Formule di quadratura approssimate 141
6.2 Formule di Newton-Cotes 142
6.3 Integrazione automatica 151

7 Problemi di Cauchy **157**
7.1 Cenni teorici 157
7.2 Algoritmi per la risoluzione 158
7.3 Metodi multipasso 164
7.4 Metodi Runge-Kutta 171
7.5 Analisi di stabilità lineare o assoluta stabilità 174

Crediti **179**
Fonti dei testi 179

1

Aritmetica floating point

1.1 Introduzione all'aritmetica floating point

1.1.1 Quattro parole chiave dell'aritmetica floating-point

Stabilità dell'algoritmo Un algoritmo è l'implementazione del metodo che vogliamo usare per risolvere il problema matematico che vogliamo trattare. La stabilità è in corrispondenza con l'errore algoritmico.

Condizionamento di un problema A cui viene associato l'errore inerente (tipico del problema).

Complessità computazionale Riguarda il tempo impiegato da un algoritmo per ottenere la soluzione. È relazionata con il tempo di calcolo e con l'errore algoritmico.

Convergenza Associata a errore analitico o di approssimazione.

1.1.2 Stabilità di un algoritmo

Un algoritmo è una sequenza finita di operazioni. Per uno stesso problema possono esistere diversi algoritmi che portano alla soluzione. Su un calcolatore, algoritmi diversi portano però a risultati diversi. L'ordine con cui vengono svolte le operazioni può portare a risultati leggermente diversi.

Si definiscono **algoritmi numericamente instabili** quelli in cui si ha un'elevata propagazione dell'errore, mentre si definiscono **algoritmi numericamente stabili** quelli in cui la propagazione dell'errore è contenuta. 16 è inoltre il numero di cifre decimali rappresentabili con Matlab.

Esempio

Vogliamo calcolare una quantità $b = 1 + a - 1$ (in aritmetica esatta $b = a$).

Algoritmo 1: $(a + 1) - 1$

Algoritmo 2: $(1 - 1) + a$

Quando a è positivo ma molto piccolo, il secondo procedimento è stabile mentre l'altro no, iniziano a esserci problemi per numeri dell'ordine di 10^{-16}.

Esempio

Consideriamo la risoluzione del sistema lineare

$$A\boldsymbol{x} = \boldsymbol{b}$$

con A matrice quadrata $n \times n$. Il sistema si può risolvere con il metodo di Kramer: la soluzione x_j del sistema può essere calcolata come il rapporto tra due determinanti

$$\frac{\det A_j}{\det A}$$

dove, indicando con A^i le colonne della matrice,

$$A = [A^1, A^2, \dots, A^n]$$

e

$$A^j = [A^1, A^{j-1}, \boldsymbol{b}, A^{j+1}, A^n$$

dove sostituiamo alla j-esima colonna il termine noto. Il determinante si calcola con la formula dei cofattori. Chiamiamo c_n il numero di operazioni moltiplicative per calcolare il determinante di una matrice A di dimensione n.

Il calcolo del determinante è ridotto al calcolo di n determinanti di matrici più piccole $(n-1) \times (n-1))$ moltiplicate per coefficienti. Quindi $c_n = n * C_{n-1} + n$ (operazione ricorsiva).

$$C_2 = 2$$

$$C_n \geq n * c_{n-1} \geq n * (n-1) * c_{n-2} \geq \cdots \geq n * (n-1) * \cdots * 3 * C_2 \geq 2n! \geq n!$$

Per trovare tutte le soluzioni del sistema dobbiamo calcolare $n+1$ determinanti di matrici di dimensione n, quindi il costo del metodo di Kramer è maggiore uguale di $(n+1)n! = (n+1)!$ e richiede tempi molto lunghi. Il costo del metodo di Gauss è invece maggiore o uguale di $n^3/3$.

Il metodo di Gauss trasforma il sistema in un sistema equivalente della forma $U(\boldsymbol{x}) = \boldsymbol{c}$ con U matrice triangolare superiore.

Calcolo un coefficiente m_k detto moltiplicatore, tale che $m_k = \frac{A_{ik}}{A_{kk}}$ al passo $k-1$.

Nell'algoritmo si combinano linearmente le due equazioni.

L'elemento a_{kk} è detto elemento pivetale. Se $a_{kk} = 0$ in aritmetica esatta si ha un problema, perché non si può dividere per 0. In aritmetica floating-point quando a_{kk} è piccolo sorgono dei problemi.

Con il calcolatore si fa in modo che l'a_{kk} che viene usato sia il meno piccolo possibile. Applichiamo la tecnica del pivet parziale: cerco l'indice di equazione $\bar{i}$ tale che a $a_{\bar{i}k}$ in modulo è il massimo per $j = k, \dots, n$ dei valori assoluti di a_{jk}.

Prima di proseguire si scambia la riga k-esima con la $\bar{i}$-esima.

Il risultato di questa tecnica è che $m_{ik} = \frac{a_{ik}}{a_{kk}} < 1$

1.1.3 Condizionamento

Esistono problemi in cui si ha un errore elevato indipendentemente dal metodo utilizzato per calcolare la soluzione. La difficoltà è una proprietà intrinseca del problema stesso.

◊ **Definizione** Si dice **problema mal condizionato** un problema in cui piccole variazioni dei dati inducono grandi variazioni nei risultati (dipendenza continua dai dati).

Esempio 1.1

Supponiamo di avere il sistema

$$\begin{cases} x + y = 2 \\ 1001x + 1000y = 2001 \end{cases}$$

e le due rette si intersecano nel punto $1x = y = 1$.

Presa la prima equazione, se perturbiamo il coefficiente di x ponendolo uguale a $1 + 1/100$ otteniamo:

$$\begin{cases} (1 + 1/100)x + y = 2 \\ 1001x + 1000y = 2001 \end{cases}$$

e il punto di intersezione è

$$\bar{x} = 1/9,\ \bar{y} = \frac{1901}{900},$$

cioè una piccola variazione dei dati da una grande variazione nella soluzione (questo avviene perché le due rette sono quasi parallele).

1.1.4 Convergenza

Si ha a che fare con la convergenza in due casi.

1. **Processi di discretizzazione**. Consideriamo ad esempio una funzione di classe almeno C^2, e consideriamo

$$f(x+h) = f(x) + hf'(x) + h^2/2f^{(2)}(\xi),\ x < \xi < x + h, \quad \text{formula di Taylor}$$

$$f'(x) = \frac{f(x+h) - f(x)}{h} - h/2f^{(2)}(\xi).$$

La discretizzazione ha un errore che si comporta come $h/2f^{(2)}(\xi)$, e va a 0 come h.

2. **Metodo iterativo**: ad esempio, vogliamo approssimare la soluzione dell'equazione $x = -\cos x$. Definisco una successione $x_{i+1} = \cos x_i$ con $i > 0$, assegnando come valore iniziale $x_0 = 1$, e proseguendo così.

1.2 Cos'è l'aritmetica floating point

I calcolatori usano la rappresentazione dei numeri in base 2, e usano la rappresentazione dei numeri in virgola mobile normalizzata.

1.2.1 Rappresentazione in virgola fissa in base 10

Consideriamo un qualsiasi numero reale $\alpha \in \mathbb{R}$, allora

$$\alpha = \text{sgn}\alpha, \alpha_n \alpha_{n-1} \dots \alpha_0 \alpha_{-1} \alpha_{-2} \alpha_{-n}$$

dove $\alpha_k \in [0, 9]$, $k \in \mathbb{Z}$, $k >= 0 =$ parte intera, $k < 0 =$ mantissa), e $\alpha_n \neq 0$.

Equivalentemente, si scrive

$$\alpha = \text{sgn}\alpha(\alpha_n 10^n + \alpha_{n-1} 10^{n-1} + \dots + \alpha_0 * 10^0 + \alpha_{-1} * 10^{-1} + \dots + \alpha_{-n} * 10^{-n} + \dots)$$

$$= \text{sgn}\alpha * (\sum_{k=-\infty}^{n} \alpha_k 10^k) \quad \text{rappresentazione}*$$

Teorema 1.1

In base β, se supponiamo che non esista $\bar{k}$ tale che $\forall k \leq \bar{k}$ $a_k = \beta - 1$, allora esiste unica la rappresentazione $*$.

Esempio 1.2

Supponiamo di considerare

$$\alpha = 0.999999\dots$$

violando l'ipotesi del teorema. Infatti, esiste $\bar{k} = -1$ tale che per ogni $k \leq \bar{k} = -1$, si ha $a_k = a_{k-1} = \beta - 1$.

$$\alpha = \sum_{k=-\infty}^{-1} a_k \beta^k$$

con il cambio di variabile $s = -k$, e tenendo conto che $\alpha_k = \beta - 1$ per ogni $k < -1$,

$$\alpha = \sum_{s=1}^{+\infty} (\beta - 1)\beta^{-s}$$

$$= (\beta - 1) * \beta^{-1} * \sum_{s=1}^{\infty} \beta^{-s+1}$$

$$= (\beta - 1) * \beta^{-1} * \sum_{s=1}^{\infty} (1/\beta)^{s-1}$$

$$= (\beta - 1) * \beta^{-1} * \sum_{k=0}^{\infty} (1/\beta)^k$$

e questa è la serie geometrica di ragione $1/\beta < 1$, quindi

$$\alpha = (\beta - 1) * \beta^{-1} * \frac{1}{1 - \beta^{-1}} =$$

$$\alpha = (\beta - 1) * \beta^{-1} * \frac{\beta}{\beta - 1} = 1$$

e abbiamo trovato una rappresentazione differente del numero.

Il vantaggio della rappresentazione in virgola fissa è la semplicità. Lo svantaggio è l'uniformità. Fissiamo il numero di cifre della parte intera del numero e della sua parte decimale. I numeri che rappresentiamo sono equispaziati (uniformità). C'è quindi uno svantaggio se vogliamo lavorare contemporaneamente con numeri grandi e piccoli.

1.2.2 Rappresentazione equivalente in virgola mobile

$$\alpha = \text{sgn}\alpha(\sum_{k=n}^{-\infty} \alpha_k \beta^k), \quad \alpha_n \neq 0$$

Raccogliendo β^{n+1} otteniamo:

$$\alpha = \text{sgn}\alpha * [\beta^{n+1} * \sum_{k=n}^{-\infty} \alpha_k \beta^{k-(n+1)}]$$

Si mette in evidenza l'ordine di grandezza del numero.
Ad esempio,

$$-1234,56 = -10^4 * 0,123456$$

Cambio di variabile: $s = k - (n + 1)$

$$\alpha = \text{sgn}\alpha * \beta^{n+1} * (\sum_{s=-\infty}^{-1} \alpha_{s+n+1} \beta^s)$$

$$= \text{sgn}\alpha * \beta^{n+1} * (\sum_{k=1}^{\infty} \alpha^{-k+n+1} \beta^{-k})$$

$$= \text{sgn}\alpha * (\beta^{n+1}) * 0.\alpha_n \alpha_{n-1} \alpha_1 \dots \alpha_0 \dots)$$

e stiamo usando una rappresentazione in virgola mobile che mette in evidenza l'ordine di grandezza.

Il vantaggio è che l'ordine di grandezza del numero viene messo subito in evidenza. Inoltre, se abbiamo un numero fissato di cifre di mantissa, allora i numeri non sono più equispaziati.

1.2.3 Struttura dell'insieme F dei numeri macchina

Sono fissati β, il numero t di cifre di mantissa, L, U che rappresentano il minimo e massimo esponente della base. Allora $\mathcal{F} = \mathcal{F}(\beta, t, L, U)$ è

$$\{x \, t.c. \, x = \text{sgn}x.(\alpha_1 \dots \alpha_t) * \beta^p\}$$

dove per le cifre vale che $0 < \alpha_i < \beta - 1$, e dove $\alpha_1 \neq 0$ e dove $L < p < U$.

Si pone $\alpha_1 \neq 0$ per risparmiare cifre, perché ad esempio, invece di scrivere $0,001$ si scrive $0,1 * 10^{-2}$.

Consideriamo per semplicità $\beta = 10$, $t = 2$, $L = -1$, $U = 1$. Analizzo quindi la struttura di $F = F(10, 2, -1, 1)$.

$$\{x = (\text{sgn}x.\alpha_1\alpha_2)10^p, \quad 0 < \alpha_1 \leq 9,\ 0 \leq \alpha_2 \leq 9\}$$

Ci si chiede qual è il più piccolo numero positivo m che si riesce a rappresentare.

$$m = 0.10 * 10^{-1} = 1/100$$

mentre

$$M = 0.99 * 10^1 = 9,9\text{cifre ed esponente al valore massimo}$$

Sulla retta rappresentiamo $0, m, M$, e i numeri prima di m e oltre M non possono essere rappresentati in F.

$$m = 0.10 * 10^{-1} = 0.01$$

$$x = 0.10 * 10^{-1},\ x_1 = 0.11 * 10^{-1}, x_2 = 0.12 * 10^{-1} \ldots, x_n = 0.20 * 10^{-1}$$

Calcolo la differenza tra i due numeri: chiamiamo *spacing* la differenza tra un numero x e il successivo, cioè, in questo caso, $x_i - x = 10^{-3}$.

Lo stesso si ripete in tutti gli intervalli della forma $(0.01, 0.02)$ oppure $(0,02, 0,03)$ etc.

Per i numeri successivi a $0.1 * 10^0$ si ha un cambiamento. Infatti, tra numeri del tipo

$$x = 0.10 * 10^0,\ x_1 = 0,11 * 10^0,\ \ldots$$

Lo *spacing* è 10^{-2}.

Considerando la rappresentazione in virgola mobile, si ha che quando i numeri sono piccoli sono più vicini fra loro, man mano che il loro valore assoluto aumenta i numeri si diradano.

L'equispaziatura è uguale per valori con lo stesso ordine di grandezza.

Consideriamo il caso generale $f = f(\beta, t, L, U)$. Allora si ha

1. $m = 0.100000000 \cdots * \beta^L$
2. $M = \beta^U * \sum_{i=1}^{t} (\beta - 1) * \beta^{-i}$.

$$M = \beta^U * \sum_{i=1}^{t} (\beta - 1) * \beta^{-i}.$$

$$M = \beta^U * \beta^{-1} * \sum_{i=1}^{t} (\beta - 1) * \beta^{-i+1}.$$

Cambio di variabile: $k + 1 = i$

$$M = \beta^U * \beta^{-1} * \sum_{k=0}^{t-1} (\beta - 1) * \beta^{-k}.$$

e per la somma della serie geometrica:

$$M = \beta^U * \beta^{-1} * (\beta - 1) * \frac{1 - \beta^{-t}}{1 - \beta^{-1}}.$$

$$M = \beta^U * (\beta - 1) * \frac{1 - \beta^{-t}}{\beta - 1}.$$

$$M = \beta^U * (1 - \beta^{-t})$$

3. Quantità di numeri positivi tra m e M (lo stesso vale per i negativi). Abbiamo t cifre di mantissa, e le possibilità con cui possiamo sceglierle sono:

$$\beta^t - \beta^{t-1}$$

Considerando gli esponenti a disposizione:

$$(U - L + 1)(\beta^t - \beta^{t-1})$$

4. *Spacing*: se poniamo $x = 0.1 * \beta^{L+1}$, il suo successivo è $0.1000001 * \beta^{L+1}$, e lo *spacing* è β^{L+1-t} nell'intervallo $(0.1 * \beta^{L+1}, 0.1 * \beta^{L+2}) = (\beta^L, \beta^{L+1})$. In generale, in un intervallo (β^p, β^{p+1}), poniamo $x = 0,1 * \beta^{p+1}$, e il successivo $0.100001 * \beta^{p+1}$, quindi lo *spacing* è $x_s - x = 0.0001 * \beta^{p+1} = \beta^{p+1-t}$ nell'intervallo (β^p, β^{p+1}) (da ricordare).

Osservazione 1.1

Lo *spacing* diventa 1 quando $\beta^{p+1-t} = 1$, ovvero $p + 1 - t = 0$, $p = t - 1$. Quindi, nell'intervallo (β^{t-1}, β^t) lo *spacing* è uguale a 1 e ci sono solo numeri interi.

1.2.4 Singola precisione

Per rappresentare un numero in singola precisione si hanno a disposizione 32 bit = 4 byte in base allo Standard i754, fissato nel 1985. In base 2, dei 32 bit, se ne usa 1 per il segno ($+ = 0$ o $- = 1$), 8 bit per l'esponente, 23 bit per la mantissa. Per evitare di utilizzare un bit per il segno dell'esponente, si considerano solo esponenti positivi.

In particolare,

$$L = -126, U = 127$$

$$-126 + 127 = 1$$

L'esponente -126 viene rappresentato con 1, e tutti gli esponenti vengono traslati di +127, e l'esponente massimo viene rappresentato con 254.

$-127 \mapsto 0$ è l'esponente che viene riservato allo zero, o per i numeri denormalizzati.

$$128 = 255 = 11111111$$

è riservato a infinito o al *not a number* (risultato di un'operazione non definita).

Uso 23 bit di mantissa, ma si ha un bit extra perché il primo bit è per forza uguale a 1 e non lo memorizzo.

Quindi, in singola precisione, $t = 24$, $m = +.10 * 2^{-126} = 1.1755 * 10^{-38}$, $M = +, 11111 * 2^{127} = 3.4028 * 10^{38}$.

1.2.5 Doppia precisione

Ogni numero occupa 64 bit e si utilizzano:

1. 1 bit per il segno del numero
2. 11 bit riservati all'esponente (shiftato di 1023)
3. 52 cifre riservate alla mantissa, + un bit extra, quindi $t = 53$.

$$m = .1 * 2^{-1022} = 2.251 * 10^{-308},\ M = .111 * 2^{1023} = 1.7977 * 10^{308}$$

realmin e *realmax* indicano la barriera di *underflow* e la barriera di *overflow*, e nella doppia precisione ci sono 16 cifre decimali.

1.3 Rappresentazione floating dei numeri

1.3.1 Riepilogo

$\mathcal{F}(\beta, t, L, U)$ è definito come $\{x = (\text{sgn}x.\alpha_1\alpha_2 \dots \alpha_t)\beta^p,\ 0 \leq \alpha_t \leq \beta - 1, \alpha_1 \neq 0, L \leq p \leq U\}$ Matlab di default lavora con la doppia precisione.

Rappresentando i numeri macchina sulla retta reale, esistono m (barriera di *underflow*) e M (barriera di *overflow*) tali che al di fuori di (m, M) i numeri non siano rappresentabili. All'aumentare dei numeri, i numeri si diradano. Lo *spacing* nell'intervallo (β^p, β^{p+1}) è β^{p+1-t}.

1.3.2 Floating di un numero

◊ **Definizione** Supponiamo di avere $x \in \mathbb{R}$ tale che

$$x = (\text{sgn}x.\alpha_1\alpha_2 \dots \alpha_t)$$

Chiamiamo ***floating*** di x, $\text{fl}(x)$ il numero in $\mathcal{F}(\beta, t, L, U)$ che associo a v, cioè il numero sul calcolatore con cui vogliamo rappresentarlo.

Ci sono varie possibilità:

1. se $p \notin (L, U)$, e in particolare se $p < L$, si ha $\text{fl}(x) = 0$ (numero troppo piccolo per essere rappresentato);
2. se invece $p > U$, allora non viene rappresentato, ci troviamo nella bariera di *overflow* e viene dato un messaggio d'errore;
3. quando $L < p < U$, e x ha più cifre di mantissa di quelle che abbiamo a disposizione, allora rappresento $\text{fl}(x) = \text{sgn}x(\alpha_1\alpha_{t-1}\alpha_{\bar{t}})\beta^p$ dove $\alpha_{\bar{t}}$ viene determinata in base al procedimento del *rounding*.

1.3.3 Procedura del *rounding*

Nel caso $L < p < U$, il calcolatore applica la procedura del *rounding*:

1. se $\alpha_{t+1} < \beta/2$, si ha $\alpha_{\bar{t}} = \alpha_t$ (*rounding* per difetto);
2. se $\alpha_{t+1} > \beta/2$, si ha $\alpha_{\bar{t}} = \alpha_t + 1$ (*rounding* per eccesso);
3. se $\alpha_{t+1} = \beta/2$, e $\alpha_k = 0\, \forall k > t + 1$ si applica il rounding to even, cioè si pone $\alpha_{\bar{t}} = \alpha_t$ se α_t è pari, altrimenti $\alpha_{\bar{t}} = \alpha_t + 1$. Il numero da rappresentare ha distanza uguale dai due estremi.

Esempio 1.3

Supponiamo di avere $\mathcal{F}(10, 2, -1, 1)$. Scrivere il *floating* dei seguenti numeri:

1. $x_1 = 5.23 = 0.523 * 10^1$ tre cifre di mantissa

$$a_{t+1} = 3 < 10/2$$

$$\text{fl}(x_1) = 0.52 * 10^1 \text{ rounding per difetto}$$

2. $x_2 = 5.27 = 0.527 * 10^1$

$$a_{t+1} = 7 > 10/2$$

$$\text{fl}(x_2) = 0.53 * 10^1 \text{ rounding per effetto}$$

3. $x_3 = 5.25 = 0.525 * 10^1$

$$a_{t+1} = 5,\ a_{t+n} = 0$$

$$\text{fl}(x_3) = 0,52 \text{ rounding to even } \textit{per difetto}$$

4. $x_4 = 5.35 = 0.535 * 10^1$

$$\text{fl}(x_4) = 0.54 * 10^1 \quad \text{rounding to even } \textit{per eccesso}$$

5. $x_5 = 0.5253 * 10^1$

è spostato più a destra dell'intervallo, quindi si approssima per eccesso.

$$\text{fl}(x_5) = 0.53 * 10^1 \text{ rappresentazione per eccesso}$$

1.3.4 Errore di rappresentazione

Dato un generico numero reale, lo si rappresenta sul calcolatore come $\text{fl}(x)$, e ci si chiede di quanto è l'errore di rappresentazione.

$\Diamond$ **Definizione** L'errore assoluto E_a è uguale a $|\text{fl}(x) - x|$. L'errore relativo E_r è uguale a $\frac{|\text{fl}(x)-x|}{|x|}$, con $x \neq 0$. Viene chiamato errore percentuale $E_p = E_r * 100$.

Esempio 1.4

Consideriamo

$$\alpha = 0.1000$$

$$\bar{\alpha} = 0.1001$$

(differiscono sulla quarta cifra decimale)

$$E_a = |\bar{\alpha} - \alpha| = 0.1 * 10^{-4}$$

$$e_r = \frac{10^{-4}}{10^{-1}} = 10^{-3}$$

$$e_p = 10^{-1} = 0.1\%$$

Consideriamo poi

$$\alpha = 1000,\ \bar{\alpha} = 1001$$

$$E_a = \alpha - \bar{\alpha} = 1$$

(l'errore assoluto differisce molto da quello precedente)

$$e_r = 1/10^3 = 10^{-3}$$

(l'errore relativo è uguale a prima).

L'errore assoluto valuta il numero e l'ordine di grandezza. L'errore relativo non considera l'ordine di grandezza dei numeri confrontati.

Teorema 1.2

Supponiamo di essere nell'intervallo (β^{p-1}, β^p), dove lo *spacing* è di β^{p-t}; consideriamo $x \in \mathbb{R}$, di cui bisogna fare il *floating*. Allora valgono le due seguenti affermazioni:

1. $E_a \leq 1/2\beta^{p-t}$ (l'errore assoluto $(x - \mathrm{fl}(x))$ è minore di metà dello *spacing*)
2. $E_r < 1/2\beta^{1-t} = u$ (l'errore relativo è strettamente minore della *round of unit*).

Dal teorema emerge che l'errore assoluto dipende anche dall'ordine di grandezza del numero (p), e aumenta in corrispondenza della grandezza del numero. L'errore relativo dipende solo dal numero di cifre di mantissa (t).

Dimostrazione (Secondo punto)

Dimostriamo il punto 2.

$$|x| = (.\alpha_1\alpha_2 \ldots \alpha_t)\beta^p.$$

$\alpha_1 \geq 1$, allora

$$|x| > (0.1) * \beta^p = \beta^{p-1}$$

$$\frac{1}{|x|} < \beta^{1-p}$$

$$E_r = \frac{|\mathrm{fl}(x) - x|}{|x|} \leq \beta^{1-p} * |\mathrm{fl}(x) - x|$$

e siccome $|\mathrm{fl}(x) - x| = E_a \leq 1/2\beta^{p-t}$ allora proseguendo con le disuguaglianze:

$$E_r \leq \beta^{1-p} * |\mathrm{fl}(x) - x| \leq 1/2\beta^{1-p}\beta^{p-t} \leq 1/2\beta^{1-t} \quad \text{disuguaglianza}*$$

Per poter dimostrare la disuguaglianza stretta (primo punto del teorema), notiamo la seguente osservazione:

Osservazione 1.2

Supponiamo di essere in base 2 con la doppia precisione, l'errore relativo che commettiamo rappresentando un numero con il suo *floating* è di 2^{-52}.

Dimostrazione (Primo punto)

Nella disuguaglianza $*$ vale l'uguale nel caso in cui $E_a = 1/2\beta^{p-t}$ (errore assoluto pari a metà dello *spacing*). Allora $\alpha_{t+1} = \beta/2$, e tutte le cifre successive $\alpha_{t+2}, \alpha_{t+3} \dots$ sono nulle. Approssimiamo x al pari più vicino (*rounding to even*).

$$|x| = (.\alpha_1\alpha_2\beta/2000)\beta^p$$

dove $\beta/2 = \alpha_{t+1}$, ma allora $\alpha_1 \neq 0$, quindi

$$x \geq (0.\alpha_1 00\beta/200)\beta^p$$

allora questo numero è strettamente maggiore di $(0.\alpha_1 000)\beta^p > \beta^{p-1}$, quindi la prima disuguaglianza nella catena sopra è stretta.

L'errore di rappresentazione in aritmetica *floating-point* è sempre più piccolo della *round of unit*.

cvd

Definizione 1.4

Supponiamo che $\bar{x}$ approssima il numero x con $E_r < \beta^{1-t}$, allora si dice che t cifre della rappresentazione di $\bar{x}$ sono esatte.

Se $\bar{x} \in \mathcal{F}$ approssima x in modo tale che le prime t cifre coincidono, allora $E_r < \beta^{1-t}$. Non vale viceversa: non è vero che se l'errore è più piccolo della quantità β^{1-t}, allora le prime t cifre coincidono.

Esempio 1.5

Consideriamo $\mathcal{F} = \mathcal{F}(10, 5, L, U)$

$$x = 0.999995$$

Siamo nel caso del *rounding to even*.

$$\mathrm{fl}(x) = 0.1 * 10^1$$

1.4 Operazioni aritmetiche con il calcolatore

Non è detto che i risultati delle operazioni tra numeri macchina siano ancora dei numeri macchina. Ci sono dei bit di guardia che permettono di rappresentare l'operazione in maniera quasi esatta, e poi si converte il risultato in numero macchina.

1.4.1 Somma

Per eseguire la somma algebrica $x \pm y$:

1. rappresentiamo x e y con i loro *floating*, cioè $\mathrm{fl}(x) + \mathrm{fl}(y)$,
2. dopo aver sommato facciamo il *floating* del risultato

cioè

$$x \pm y = \text{fl}(\text{fl}(x) + \text{fl}(y))$$

L'errore relativo $\frac{\text{fl}(x)-x}{x} = \varepsilon$, e si pone $\text{fl}(x) = x(1+\varepsilon)$. ε non un infinitesimo.

$$\text{fl}(x)+\text{fl}(y) = x(1+\varepsilon_x)+y(1+\varepsilon_y) = \text{fl}(\text{fl}(x)+\text{fl}(y)) = (1+\varepsilon_s)(x(1+\varepsilon_x)+y(1+\varepsilon_y))$$

Al primo ordine, si ha

$$= x(1+\varepsilon_x) \pm y(1+\varepsilon_y) + (x \pm y)\varepsilon_s$$

$$= (x \pm y) + x\varepsilon_x + y\varepsilon_y + (x \pm y)\varepsilon_s$$

Considerando l'errore relativo al primo ordine:

$$E_r = |\frac{\text{fl}(x \pm y) - (x \pm y)}{x \pm y}| =^{\circ} \frac{|(x \pm y) + x\varepsilon_x + y\varepsilon_y + (x+y)\varepsilon_s - (x \pm y)|}{x \pm y}$$

$$=^{\circ} \frac{x\varepsilon_x + y\varepsilon_y + (x+y)\varepsilon_s|}{x \pm y}$$

$$= |\frac{x}{x \pm y} * \varepsilon_x \pm \frac{y}{x \pm y}\varepsilon_y + \varepsilon_s|$$

$$\leq |\frac{x}{x \pm y}| * |\varepsilon_x| \pm |\frac{y}{x \pm y}| * |\varepsilon_y| + |\varepsilon_s|$$

e in base al teorema sulla *round of unit*

$$\leq u * (\frac{|x|}{|x \pm y|} + \frac{|y|}{|x \pm y|} + 1)$$

Distinguiamo due casi.

1. Se i due numeri hanno segno concorde, allora $x/|x+y|$ e $y/|x+y|$ sono frazioni proprie, e sono minori o uguali ad 1. Allora l'errore relativo è maggiorato al primo ordine da $3u = 3 * 2^{-52}$. L'operazione è "stabile", e non dipende dai dati.
2. Nel caso di segno discorde, l'errore dipende dai dati, e non è sempre "stabile", perché può capitare il fenomeno della cancellazione numerica.

Esempio 1.6

Supponiamo $\mathcal{F} = \mathcal{F}(10, 5, -6, 6)$. Consideriamo

$$x = 0.123456,\ y = -0.123454$$

$$\text{fl}(x) = 0.12346 * 10^0 \text{ per eccesso}$$

$$\text{fl}(y) = -0.12345 * 10^0 \text{ tper difetto}$$

In aritmetica esatta

$$x + y = 0.2 * 10^{-5}$$

Per il calcolatore

$$\mathrm{fl}(x) + \mathrm{fl}(y) = 0.1 * 10^{-4} \in \mathcal{F}$$

Non c'è nessuna cifra esatta.

$$e_A = 0.8 * 10^{-5},\ e_r = 4,\ e_p = 400\%$$

La somma è fra due numeri quasi uguali in valore assoluto e con segno opposto: si ha un errore molto elevato dovuto all'approssimazione. Questo fenomeno viene chiamato **cancellazione numerica**.

L'errore relativo totale è fatto al primo ordine da tre termini:

1. ε_s, errore locale generato nell'operazione;
2. errore relativo al dato x moltiplicato per un coefficiente c_x, ed è $\frac{x}{x \pm y}$;
3. errore relativo al dato y, moltiplicato per c_y

Allora $\varepsilon_x, \varepsilon_y$ danno l'errore di rappresentazione sui dati, mentre c_x, c_y, detti coefficienti di amplificazione.

L'errore totale al primo ordine è

$$\varepsilon_s + c_x * \varepsilon_x + c_y * \varepsilon_y$$

(coefficiente lungo il ramo moltiplicato per errore)

Esercizio 1.1

Se l'operazione si scrive come $f(x, y)$ allora

$$c_x = \frac{x}{f(x,y)} * \frac{\partial f}{\partial x}$$

$$c_y = \frac{y}{f(x,y)} * \frac{\partial f}{\partial y}$$

Verificare che questo è vero per i risultati ottenuti per la somma, e dimostrarlo nel caso di moltiplicazione e divisione.

Nel caso della somma:

$$\begin{aligned}
&f(x,y) = x + y \\
&\frac{\partial f}{\partial x} = 1 \\
&\frac{\partial f}{\partial y} = 1 \\
&\longrightarrow c_x = \frac{|x|}{|x+y|} * 1 \\
&\longrightarrow c_y = \frac{|y|}{|x \pm y|}
\end{aligned}$$

quindi

$$E_r = \frac{x}{|x \pm y|} * \varepsilon_x + \frac{y}{|x \pm y|} * \varepsilon_y + \varepsilon_s \quad \text{coincide con l'espressione ottenuta}$$

Nel caso della moltiplicazione ricavo il risultato con i due procedimenti:
Procedimento 1: $x * y$ viene eseguita dal calcolatore come

$$\text{fl}(\text{fl}(x) * \text{fl}(y))$$

e considerando che $\text{fl}(x) = x(1 + \varepsilon_x)$ e $\text{fl}(y) = y * (1 - \varepsilon_y)$ otteniamo:

$$= \text{fl}(x(1 + \varepsilon_x) * y * (1 + \varepsilon_y))$$

$$= (1 + \varepsilon_m)[xy(1 + \varepsilon_x)(1 + \varepsilon_y)]$$

e al primo ordine otteniamo:

$$= xy + xy\varepsilon_x + xy\varepsilon_y + xy\varepsilon_m$$

e l'errore relativo è:

$$E_r = \frac{|\text{fl}(xy) - xy|}{|xy|} =$$

$$=^{\circ} \frac{xy + xy\varepsilon_x + xy\varepsilon_y + xy\varepsilon_m - xy}{|xy|}$$

$$=^{\circ} \frac{xy\varepsilon_x + xy\varepsilon_y + xy\varepsilon_m}{|xy|}$$

$$= \varepsilon_x + \varepsilon_y + \varepsilon_m$$

Procedimento 2:

$$\begin{aligned} f(x, y) &= xy \\ \frac{\partial f}{\partial x} &= y \\ \frac{\partial f}{\partial y} &= x \\ \longrightarrow c_x &= \frac{x}{|xy|} * y = 1 \\ \longrightarrow c_y &= \frac{y}{|xy|} * x = 1 \end{aligned}$$

quindi

$$E_r = c_x\varepsilon_x + c_y\varepsilon_y + \varepsilon_m$$

$$= \varepsilon_x + \varepsilon_y + \varepsilon_m$$

1.4.2 Derivazione dei coefficienti di amplificazione

Vogliamo calcolare $f(x_1, x_2, \ldots, x_n)$. Ipotizziamo che f sia sufficientemente regolare, e che non ci sia errore nel calcolo del valore di f, ma solo nella rappresentazione dei dati.

Chiamiamo $x = (x_1, \ldots, x_n)$, e ne consideriamo il *floating*, cioè $\mathrm{fl}(x) = (\mathrm{fl}(x_1), \ldots, \mathrm{fl}(x_n))$.

$$E_r = \frac{f(\mathrm{fl}(x)) - f(x)}{f(x)}$$

Caso 1: $n = 1$

$$E_r = \frac{f(x) + f'(x) * (\mathrm{fl}(x) - x) + o(\mathrm{fl}(x) - x) - f(x)}{f(x)}$$

Considerando $\frac{f'(x)*(\mathrm{fl}(x)-x)}{x}$, moltiplichiamo numeratore e denominatore per v:

$$\frac{x * f'(x) * (\mathrm{fl}(x) - x)}{x * f(x)}$$

$$= \frac{x * f'(x)}{f(x)} * \frac{\mathrm{fl}(x) - x}{x} + \frac{o(\mathrm{fl}(x) - x)}{f(x)}$$

$$= (c_x * \varepsilon_x) + o(\mathrm{fl}(x) - x)$$

Caso 2: n generico

$$E_r = \frac{1}{f(x)}[f(x) + \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}(x) * (\mathrm{fl}(x_i) - x_i) + o(|\mathrm{fl}(x) - x|) - f(x)]$$

$$E_r = [\sum_{i=1}^{n} \frac{x_i}{f(x} * \frac{\partial f}{\partial x_i}(x)] * \frac{\mathrm{fl}(x_i) - x_i}{x_i} + \frac{o(|\mathrm{fl}(x) - x|)}{f(x)}$$

$$= \sum_i c_{x_i} \varepsilon_{x_i} + o(\mathrm{fl}(x) - x)$$

1.4.3 Divisione

$$x/y = \frac{x(1 + \varepsilon_x)}{y(1 + \varepsilon_y)}$$

$$(1 + \varepsilon_d)\frac{x(1 + \varepsilon_x)}{y(1 + \varepsilon_y)}$$

$$= x/y * \frac{(1 + \varepsilon_x)(1 + \varepsilon_d)}{1 + \varepsilon_y}$$

Sappiamo che

$$\frac{1}{1 + \varepsilon_y} \circ 1 - \varepsilon_y$$

Quindi

$$= \frac{x}{y}(1 + \varepsilon_x)(1 - \varepsilon_y)(1 + \varepsilon_d)$$

$$= x/y * (1 + \varepsilon_x - \varepsilon_y + \varepsilon_d)$$

Considerando l'errore relativo:

$$\frac{|x/y| * (1 + \varepsilon_x - \varepsilon_y + \varepsilon_d) - |x/y|}{x/y}$$

$$1 + \varepsilon_x - \varepsilon_y + \varepsilon_d - 1$$

$$= \varepsilon_x - \varepsilon_y + \varepsilon_d \leq |\varepsilon_x| + |\varepsilon_y| + |\varepsilon_d| \leq 3u$$

e si ottiene lo stesso risultato della moltiplicazione, cioè l'operazione di divisione non dipende dai dati.

$$E_r = |c_x| * \varepsilon_x + |c_y| * \varepsilon_y + \varepsilon_d$$

e ancora $c_x = \frac{x*1/y}{x/y} = 1$, $c_y = \frac{-x/y^2*y}{x/y} = -1$.

1.5 Teoria dell'errore nella sua generalità

1.5.1 Calcolo del valore di una funzione

Le funzioni effettivamente calcolabili sono le **funzioni razionali**, cioè quelle che sono date da un numero finito di operazioni $+, -, *, /$. Le funzioni non razionali vengono approssimate in qualche modo con funzioni razionali. È data una funzione $f\colon \mathbb{R}^n \to \mathbb{R}$, e vogliamo calcolarne il valore in v.

Il calcolo che effettuo con il calcolatore è affetto da vari tipi di errore.

Errore inerente è l'errore generato dall'errore di rappresentazione sui dati. È connesso al condizionamento ed è una caratteristica della funzione da calcolare.

Errore algoritmico è generato dall'errore commesso nelle operazioni in aritmetica *floating point*. È combinazione lineare degli errori locali delle singole operazioni. È connesso alla stabilità

Errore analitico o di approssimazione è l'errore generato dall'approssimazione di f non razionale con un'opportuna funzione razionale. È connesso alla convergenza.

1.5.2 Funzioni razionali

Chiamiamo x il valore in cui vogliamo calcolare la funzione, di cui consideriamo il *floating* $\mathrm{fl}(x)$, ψ è la funzione effettivamente calcolata. Valore da calcolare: $f(x)$, valore effettivamente calcolato: $\psi(\mathrm{fl}(x))$.

Supponiamo di avere $x \neq 0$, e f funzione razionale tale che $f(x) \neq 0$ e $f(\mathrm{fl}(x)) \neq 0$. Vogliamo frammentare l'errore (relativo) totale $\frac{\psi(\mathrm{fl}(x)) - f(x)}{f(x)}$ in diversi tipi di errori:

1. errore inerente: $e_{in} = \frac{f(\mathrm{fl}(x)) - f(x)}{f(x)}$. Si assume che f sia calcolata esattamente, e si considera solo l'errore dipendente dalla rappresentazione dei dati.
2. errore algoritmico: misura l'errore generato dall'uso di ψ al posto di f, calcolate nel *floating* di v. La sua espressione è $e_{alg} = \frac{\psi(\mathrm{fl}(x)) - f(\mathrm{fl}(x))}{f(\mathrm{fl}(x))}$

Vale il seguente teorema.

Teorema 1.3

L'errore totale è uguale a $e_{alg}(1+e_{in})+e_{in}$, e al primo ordine è uguale a $e_{alg}+e_{in}$. Inoltre

$$e_{in} = \sum_{i=1}^{n} c_i \varepsilon_{x_i}, \quad , c_i = \frac{x_i}{f(x)} * \frac{\partial f}{\partial x_i}$$

con

$$|\varepsilon_{x_i}| < u$$

Dimostrazione

$$E_r = \frac{\psi(\mathrm{fl}(x)) - f(x)}{f(x)}$$

$$= \frac{\psi(\mathrm{fl}(x))}{f(x)} - 1$$

$$= \frac{\psi(\mathrm{fl}(x))}{f(\mathrm{fl}(x))} * \frac{f(\mathrm{fl}(x))}{f(x)} - 1$$

$$= (E_{alg}+1) * (e_{in}+1) - 1$$

$$e_{alg} + e_{in} + e_{alg}e_{in} + 1 - 1$$

$$= e_{alg}(1+e_{in}) + e_{in}$$

cvd

1.5.3 Caso generale funzione non razionale

Supponiamo di avere $x \neq 0$, una funzione f non razionale che viene approssimata con una funzione g razionale, e $\psi(\mathrm{fl}(x))$ è la funzione effettivamente calcolata.

$$E_r = \frac{\psi(\mathrm{fl}(x)) - f(x)}{f(x)}$$

si frammenta in:

1. errore analitico: dipende dal fatto di usare g al posto di f, e ha espressione $e_{an} = \frac{g(x)-f(x)}{f(x)}$
2. errore inerente: $e_{in} = \frac{f(\mathrm{fl}(x))-f(x)}{f(x)}$
3. errore algoritmico: $E_{alg} = \frac{\psi(\mathrm{fl}(x))-g(\mathrm{fl}(x))}{g(\mathrm{fl}(x))}$

Teorema 1.4

L'errore totale al primo ordine è dato da

$$e_{in} + e_{alg} + e_{an,\mathrm{fl}(x)}$$

con $e_{an,\mathrm{fl}(x)} = \frac{g(\mathrm{fl}(x))-f(\mathrm{fl}(x))}{f(\mathrm{fl}(x))}$

Dimostrazione

$$
\begin{aligned}
e_r &= \frac{\psi(\mathrm{fl}(x)) - f(x)}{f(x)} \\
&= \frac{\psi(\mathrm{fl}(x))}{f(x)} - 1 \\
&= \frac{\psi(\mathrm{fl}(x))}{g(\mathrm{fl}(x))} * \frac{g(\mathrm{fl}(x))}{f(x)} - 1 \\
&= (e_{alg} + 1)\frac{g(\mathrm{fl}(x))}{f(x)} - 1 \\
&= (e_{alg} + 1)[\frac{g(\mathrm{fl}(x))}{f(\mathrm{fl}(x))} * \frac{f(\mathrm{fl}(x))}{f(x)}] - 1 \\
&= (e_{alg} + 1)(e_{in} + 1)\frac{g(\mathrm{fl}(x))}{f(\mathrm{fl}(x))} - 1 \\
&= (e_{alg} + 1)(e_{in} + 1)\frac{g(\mathrm{fl}(x)) - f(\mathrm{fl}(x)) + f(\mathrm{fl}(x))}{f(\mathrm{fl}(x))} - 1 \\
&= (e_{alg} + 1)(e_{in} + 1)[\frac{g(\mathrm{fl}(x)) - f(\mathrm{fl}(x))}{f(\mathrm{fl}(x))} + 1] - 1 \\
&= (e_{alg} + 1)(e_{in} + 1)(e_{an,\mathrm{fl}(x)} + 1) - 1
\end{aligned}
$$

Al primo ordine

$$e_{alg} + e_{in} + e_{an,\mathrm{fl}(x)}$$

cvd

Se $e_{an,x}$ e $e_{an,\mathrm{fl}(x)}$ differiscono di un termine del secondo ordine, allora vale il risultato con il vero errore analitico.

1.6 Condizionamento e stabilità

1.6.1 Condizionamento

Il condizionamento fa riferimento all'errore inerente. Se $|c_{x_i}|$ sono elevati in modulo, allora a un piccolo errore ε_{x_i} sulla rappresentazione dei dati corrisponde un grande errore su $f(x)$, e il problema è mal condizionato.

Esempio 1.7

Consideriamo un'equazione del tipo

$$ax^2 + bx + c = 0, \quad \delta = b^2 - 4ac$$

Per calcolare le radici si ha la formula $x_{1,2} = \frac{-b \pm \sqrt{\delta}}{2a}$.

Se $\sqrt{\delta} = b$, per calcolare x_1 sto sommando due quantità quasi uguali, e quindi c'è un grande errore di cancellazione. Invece il calcolo di x_2 non dà problemi.

Viceversa, se $\sqrt{\delta} = -b$, l'errore di cancellazione si ha per x_2. Allora possiamo rimodellizzare il problema: se $\sqrt{\delta} \simeq -b$, allora calcolo x_2 con la formula solita senza avere problemi, invece calcolo x_1 ricordando che

$$x_1 x_2 = c/a$$

quindi

$$x_1 = \frac{c}{a x_2}$$

e il problema torna a essere stabile, dopo essere stato rimodellizzato.

1.6.2 Stabilità

La stabilità è connessa all'errore algoritmico, e riguarda l'ordine esatto con cui vengono fatte le operazioni. L'errore algoritmico viene generato nella valutazione di $\psi(\mathrm{fl}(x))$.

ψ è una funzione razionale, cioè ha un'espressione analitica data da un numero finito di operazioni di somma e sottrazione. L'errore algoritmico è una combinazione lineare degli errori locali generati nelle singole operazioni.

Simbolicamente,

$$|e_{alg}| > \theta(u) + O(u^2)$$

dove θ è indipendente dalla round of unit, e tiene conto della combinazione lineare dei coefficienti.

Osservazione 1.3

Se abbiamo un problema mal condizionato, l'analisi dell'errore algoritmico è inutile, perché questo errore è trascurabile rispetto all'errore inerente.

1.6.3 Esempio di rimodellizzazione di un problema

Esercizio 1.2

Analizzare l'errore nel calcolo della funzione

$$f(x, y) = x^2 - y^2 = (x + y)(x - y).$$

Chiamiamo a_1 l'algoritmo $f = x^2 - y^2$ che è dato dai seguenti passi:

1. poniamo $z_1 = x * x$;
2. poniamo $z_2 = y * y$;
3. eseguiamo l'operazione $z_3 = z_1 - z_2$.

Nella moltiplicazione tengo conto che $c_x = c_y = 1$, mentre nella sottrazione si ha $c_x = \frac{x}{x-y}$ e $c_y = \frac{-y}{x-y}$.

Rappresento graficamente l'algoritmo:

$$\begin{array}{cccc}
x & x & y & y \\
\uparrow (1) & \uparrow (1) & \uparrow (1) & \uparrow (1) \\
 & z_1 & z_2 & \\
 & \uparrow (\frac{z_1}{z_3}) & \uparrow (\frac{-z_2}{z_3}) & \\
 & \multicolumn{2}{c}{z_3 = z_1 - z_2} &
\end{array}$$

$$\varepsilon_{z_3} + \frac{z_1}{z_3}(\varepsilon_{z_1} + \varepsilon_x + \varepsilon_x) - \frac{z_2}{z_3}(\varepsilon_{z_2} + \varepsilon_y + \varepsilon_y)$$

Vogliamo rappresentare la formula generale come somma di errore algoritmico e inerente.

$$e_r = \varepsilon_{z_3} + 2\frac{z_1}{z_3} * \varepsilon_x - 2\frac{z_2}{z_3} * \varepsilon_y + \frac{z_1}{z_3} * \varepsilon_{z_1} - \frac{z_2}{z_3} * \varepsilon_{z_2}$$

e sostituendo a z_1, z_2, z_3 le loro espressioni in termini di x, y, z si ha:

$$= \frac{2x^2}{x^2 - y^2}\varepsilon_x - \frac{2y^2}{x^2 - y^2} * \varepsilon_y + \varepsilon_{z_3} + \frac{x^2}{x^2 - y^2}\varepsilon_{z_1} - \frac{y^2}{x^2 - y^2}\varepsilon_{z_2}$$

I primi due termini rappresentano l'errore inerente, mentre gli altri tre rappresentano l'errore algoritmico. Se $x^2 - y^2$ è piccolo, il problema è mal condizionato, e l'errore inerente domina.

Esercizio 1.3

Ricavare lo stesso risultato per via analitica.

$$f(x, y) = x^2 - y^2$$

$$= \mathrm{fl}\{\mathrm{fl}[\mathrm{fl}(x) * \mathrm{fl}(x)] - \mathrm{fl}[\mathrm{fl}(y) * \mathrm{fl}(y)]\}$$

$$= \mathrm{fl}\{\mathrm{fl}[(1 + \varepsilon_x) * |x| * (1 + \varepsilon_x) * |x|] - \mathrm{fl}[(1 + \varepsilon_y) * |y| * (1 + \varepsilon_y) * |y|]\}$$

$$= \mathrm{fl}\{(1 + \varepsilon_{z_1})(1 + \varepsilon_x) * |x| * (1 + \varepsilon_x) * |x| - (1 + \varepsilon_{z_2})(1 + \varepsilon_y) * |y| * (1 + \varepsilon_y) * |y|\}$$

$$= (1 + \varepsilon_{z_3})\{(1 + \varepsilon_{z_1})(1 + \varepsilon_x) * |x| * (1 + \varepsilon_x) * |x| - (1 + \varepsilon_{z_2})(1 + \varepsilon_y) * |y| * (1 + \varepsilon_y) * |y|\}$$

$$= (1 + \varepsilon_{z_3})\{x^2(1 + \varepsilon_{z_1})(1 + \varepsilon_x)^2 - y^2(1 + \varepsilon_{z_2})(1 + \varepsilon_y)^2\}$$

e al primo ordine:

$$= (1 + \varepsilon_{z_3})\{x^2(1 + \varepsilon_{z_1})(1 + 2\varepsilon_x) - y^2(1 + \varepsilon_{z_2})(1 + 2\varepsilon_y)\}$$

$$= (1 + \varepsilon_{z_3})\{x^2(1 + \varepsilon_{z_1} + 2\varepsilon_x) - y^2(1 + \varepsilon_{z_2} + 2\varepsilon_y)\}$$

$$= x^2(1 + \varepsilon_{z_1} + 2\varepsilon_x + \varepsilon_{z_3}) - y^2(1 + \varepsilon_{z_2} + 2\varepsilon_y + \varepsilon_{z_3})$$

Valuto l'errore relativo:

$$E_r = \frac{\mathrm{fl}(x^2 - y^2) - (x^2 - y^2)}{x^2 - y^2}$$

$$= \frac{x^2(\varepsilon_{z_1} + 2\varepsilon_x + \varepsilon_{z_3}) - y^2(\varepsilon_{z_2} + 2\varepsilon_y + \varepsilon_{z_3})}{x^2 - y^2}$$

$$= \frac{(x^2 - y^2)\varepsilon_{z_3} + 2x^2\varepsilon_x + x^2\varepsilon_{z_1} - y^2\varepsilon_{z_2} - 2y^2\varepsilon_y}{x^2 - y^2}$$

$$= \frac{2x^2}{x^2 - y^2}\varepsilon_x - \frac{2y^2}{x^2 - y^2} * \varepsilon_y + \frac{x^2}{x^2 - y^2}\varepsilon_{z_1} - \frac{y^2}{x^2 + y^2}\varepsilon_{z_2} + \varepsilon_{z_3}$$

e si ottiene lo stesso risultato di prima.

Rimodellizzo il problema applicando l'algoritmo 2.

1. $w_1 = x + y$
2. $w_2 = x - y$
3. $w_3 = w_1 * w_2$

Assumiamo che stiamo facendo conti sui numeri macchina, perché l'errore inerente si può calcolare a parte, cioè poniamo $\varepsilon_x = \varepsilon_y = 0$. Graficamente otteniamo:

$$\begin{array}{cc|c|cc}
x & y & & x & y \\
\uparrow (c_x) & \uparrow (c_y) & & \uparrow (c_x) & \uparrow (c_y) \\
 & w_1 & & w_2 & \\
 & \uparrow (1) & & \uparrow (1) & \\
 & & z & &
\end{array}$$

quindi

$$e_{alg} = \varepsilon_3 + 1 * (\varepsilon_1 + c_x * \varepsilon_x) + 1 * (\varepsilon_2 + c_y * \varepsilon_y)$$

e allora, siccome $\varepsilon_x = \varepsilon_y = 0$ ottengo:

$$e_{alg} = \varepsilon_3 + 1 * (\varepsilon_1 + 0) + 1 * (\varepsilon_2 + 0)$$

$$= \varepsilon_1 + \varepsilon_2 + \varepsilon_3$$

Esercizio 1.4

Ricavare lo stesso risultato per via analitica.

$$\mathrm{fl}(x^2 - y^2) =$$

$$\mathrm{fl}[\mathrm{fl}(x(1 + \varepsilon_x) + y(1 + \varepsilon_y)) * \mathrm{fl}(x(1 + \varepsilon_x) - y(1 + \varepsilon_y))] =$$

$$(1 + \varepsilon_3)[(1 + \varepsilon_1)(x(1 + \varepsilon_x) + y(1 + \varepsilon_y)) * (1 + \varepsilon_2) * [x(1 + \varepsilon_x) - y(1 + \varepsilon_y)]] =$$

Al primo ordine

$$(1 + \varepsilon_3)[(1 + \varepsilon_1) * (1 + \varepsilon_2) * [x^2(1 + 2\varepsilon_x) - y^2(1 + 2\varepsilon_y)] =$$

$$(1 + \varepsilon_1 + \varepsilon_2 + \varepsilon_3) * [x^2(1 + 2\varepsilon_x) - y^2(1 + 2\varepsilon_y)] =$$

$$x^2(1+2\varepsilon_x+\varepsilon_1+\varepsilon_2+\varepsilon_3)-y^2(1+2\varepsilon_y+\varepsilon_1+\varepsilon_2+\varepsilon_3)$$

$$E_r=\frac{x^2(1+2\varepsilon_x+\varepsilon_1+\varepsilon_2+\varepsilon_3)-y^2(1+2\varepsilon_y+\varepsilon_1+\varepsilon_2+\varepsilon_3)-x^2+y^2}{|x^2-y^2|}$$

$$E_r=\frac{x^2(2\varepsilon_x+\varepsilon_1+\varepsilon_2+\varepsilon_3)-y^2(2\varepsilon_y+\varepsilon_1+\varepsilon_2+\varepsilon_3)}{|x^2-y^2|}$$

$$E_r=\frac{(x^2-y^2)(\varepsilon_1+\varepsilon_2+\varepsilon_3)+2x^2*\varepsilon_x-2y^2*\varepsilon_y}{|x^2-y^2|}$$

$$E_r\leq\frac{(x^2-y^2)(\varepsilon_1+\varepsilon_2+\varepsilon_3)}{x^2-y^2}+\frac{2x^2*\varepsilon_x}{x^2-y^2}+\frac{2y^2*\varepsilon_y}{|x^2-y^2|}$$

$$E_r\leq\varepsilon_1+\varepsilon_2+\varepsilon_3+\frac{2x^2*\varepsilon_x}{x^2-y^2}+\frac{2y^2*\varepsilon_y}{|x^2-y^2|}$$

Gli ultimi due addendi rappresentano l'errore inerente, uguale a quello del primo algoritmo, mentre

$$E_{alg}=\varepsilon_1+\varepsilon_2+\varepsilon_3\leq 3u$$

Nel primo algoritmo:

$$|E_{alg,1}|\leq u(1+\frac{x^2+y^2}{x^2-y^2})$$

invece

$$|E_{alg,2}|\leq 3u$$

Il secondo algoritmo è stabile e indipendente dai dati, ma quando $|x^2-y^2|$ è piccolo il problema diventa mal condizionato quindi questo vantaggio non serve a niente, tranne nel caso in cui x e y sono numeri macchina.

La cosa migliore è fare subito le operazioni rischiose (somma e sottrazione), e lasciare moltiplicazioni e divisioni per ultime, in modo che l'errore accumulato sia minore.

1.6.4 Errore nel calcolo della somma di n numeri

Calcoliamo l'errore della funzione

$$f(x)=\sum_{i=1}^{n}x_i$$

e consideriamo

$$c_i=\frac{x_i}{f(x)}*\frac{\partial f}{\partial x_i}=\frac{x_i}{f(x)}$$

$$E_{in}=\sum_{i=1}^{n}\varepsilon_{x_i}*\frac{x_i}{f(x_1,\ldots,x_n)}$$

$$= |\frac{1}{f(x)} * \sum_{i=1}^{n} x_i * \varepsilon_{x_i}|$$

$$\leq |\frac{nu}{f(x)}| * \max |x_i|$$

L'errore inerente dipende dai dati e dal loro numero.

Se siamo nella situazione in cui gli x_i hanno tutti segni concordi, allora:

$$|\sum_i x_i| = \sum_i |x_i|$$

Quindi, tenendo anche conto che $f(x) = |\sum_{i=1}^{n} x_i|$, si ha:

$$E_{in} = \frac{1}{|\sum_i x_i|} * |\sum_i x_i \varepsilon_{x_i}|$$

$$\leq \frac{1}{|\sum x_i|} * \sum_i |x_i||\varepsilon_{x_i}|$$

$$\leq \frac{1}{\sum_i |x_i|} * u * \sum_i |x_i| < u$$

Allora in questo caso il problema è ben condizionato, mentre nel caso generale l'errore inerente dipende dal valore dei dati e dal loro numero.

Rappresento il grafo dell'errore relativo nel caso della somma di quattro numeri x_1, x_2, x_3, x_4 ponendo:

1. $z_1 = x_1 + x_2$;
2. $z_2 = z_1 + x_3 = (x_1 + x_2) + x_3$;
3. $z_3 = x_4 + z_2 = x_4 + (x_1 + x_2 + x_3)$.

Il grafico è

$$\begin{array}{cccc}
 & & x_2 & x_1 \\
 & & \uparrow (c_{x_2}) & \uparrow (c_{x_1}) \\
 & x_3 & z_1 & \\
 & \uparrow (c_{x_3}) & \uparrow (c_{z_1}) & \\
x_4 & z_2 & & \\
\uparrow (c_{x_4}) & \uparrow (c_{z_2}) & & \\
z_3 & & &
\end{array}$$

Supponiamo che x_1, x_2, x_3, x_4 siano numeri macchina, quindi $\varepsilon_{x_i} = 0$, e rimane:

$$E_{alg} = \varepsilon_3 + c_{z_2} * (\varepsilon_2 + c_{z_1} * \varepsilon_1)$$

e sostituendo ai coefficienti le loro espressioni:

$$= \varepsilon_3 + \frac{z_2}{z_3} * (\varepsilon_2 + \frac{z_1}{z_2}\varepsilon_1)$$

$$= \varepsilon_3 + \frac{z_2}{z_3} * \varepsilon_2 + \frac{z_2}{z_3} * \frac{z_1}{z_2}\varepsilon_1$$

$$= \varepsilon_3 + \frac{z_2}{z_3} * \varepsilon_2 + \frac{z_1}{z_3} * \varepsilon_1$$

Sostituiamo le espressioni di z_i espressi in funzione delle x_i:

$$= \varepsilon_3 + \frac{x_1 + x_2 + x_3}{x_1 + x_2 + x_3 + x_4} * \varepsilon_2 + \frac{x_1 + x_2}{x_1 + x_2 + x_3 + x_4} * \varepsilon_1$$

allora anche l'errore algoritmico dipende dai dati.

Esercizio 1.5

Trovare l'espressione dell'errore algoritmico per via analitica.

Supponiamo che gli x_i siano numeri macchina, cioè che $\mathrm{fl}(x_i) = x_i$.

$$x_1 + x_2 + x_3 + x_4 =$$

$$\{[(x_1 + x_2) + x_3] + x_4\}$$

$$\mathrm{fl}\{\mathrm{fl}[\mathrm{fl}(x_1 + x_2) + x_3] + x_4\}$$

$$= \{[(x_1 + x_2)(1 + \varepsilon_1) + x_3](1 + \varepsilon_2) + x_4\}(1 + \varepsilon_3)$$

Eseguendo i prodotti e trascurando i termini di ordine superiore al primo:

$$= \{[x_1 + x_2 + x_3 + x_1\varepsilon_1 + x_2\varepsilon_1](1 + \varepsilon_2) + x_4\}(1 + \varepsilon_3)$$

$$= \{x_1 + x_2 + x_3 + x_4 + x_1\varepsilon_1 + x_2\varepsilon_1 + x_1\varepsilon_2 + x_2\varepsilon_2 + x_3\varepsilon_2\}(1 + \varepsilon_3)$$

$$= x_1 + x_2 + x_3 + x_4 + x_1\varepsilon_1 + x_2\varepsilon_1 + x_1\varepsilon_2 + x_2\varepsilon_2 + x_3\varepsilon_2 + x_1\varepsilon_3 + x_2\varepsilon_3 + x_3\varepsilon_3 + x_4\varepsilon_3$$

Allora

$$E_{alg} = \frac{|\mathrm{fl}(x_1 + x_2 + x_3 + x_4) - x_1 - x_2 - x_3 - x_4|}{|x_1 + x_2 + x_3 + x_4|}$$

$$= \frac{x_1 + x_2 + x_3 + x_4 + x_1\varepsilon_1 + x_2\varepsilon_1 + x_1\varepsilon_2 + x_2\varepsilon_2 + x_3\varepsilon_2 + x_1\varepsilon_3 + x_2\varepsilon_3 + x_3\varepsilon_3 + x_4\varepsilon_3 - x_1 - x_2 - x_3 - x_4}{x_1 + x_2 + x_3 + x_4}$$

$$= \frac{x_1\varepsilon_1 + x_2\varepsilon_1 + x_1\varepsilon_2 + x_2\varepsilon_2 + x_3\varepsilon_2 + x_1\varepsilon_3 + x_2\varepsilon_3 + x_3\varepsilon_3 + x_4\varepsilon_3}{x_1 + x_2 + x_3 + x_4}$$

$$= \varepsilon_3 + \frac{x_1\varepsilon_1 + x_2\varepsilon_1}{x_1 + x_2 + x_3 + x_4} + \frac{x_1\varepsilon_2 + x_2\varepsilon_2 + x_3\varepsilon_2}{x_1 + x_2 + x_3 + x_4}$$

$$= \varepsilon_3 + \frac{\varepsilon_1(x_1 + x_2)}{x_1 + x_2 + x_3 + x_4} + \frac{\varepsilon_2(x_1 + x_2 + x_3)}{x_1 + x_2 + x_3 + x_4}$$

e si ottiene il risultato precedente.

Nel caso della somma di n numeri si ottiene la formula generale:

$$E_{alg} = |\frac{1}{f(x)} \sum_{i=1}^{n-1} \sum_{j=1}^{i+1} x_j\varepsilon_i|$$

Se invece gli x_i sono di segno concorde, allora

$$\sum_{j=1}^{i+1} |x_j| \leq |f(x)|$$

quindi, applicando la disuguaglianza triangolare, si ottiene:

$$E_{alg} \leq |\frac{1}{f(x)}| \sum_{i=1}^{n-1} \sum_{j=1}^{i+1} |x_j| \varepsilon_i|$$

$$E_{alg} \leq \sum_{i=1}^{n-1} |\varepsilon_i| \leq (n-1)u$$

Allora l'errore algoritmico non dipende dai dati.

1.6.5 Ottimizzazione dell'algoritmo per la somma di n numeri

Procedimento 1: per ridurre l'errore nella somma, conviene ordinare i numeri per valore assoluto, dal più piccolo al più grande. In questo modo risulta che, se consideriamo le medie aritmetiche dei primi $i+1$ numeri, si ha:

$$|1/i * \sum_{j=1}^{i} x_j| \leq \frac{1}{i+1} * |\sum_{j=1}^{i+1} x_j| \leq \frac{1}{n-1}|f(x)|$$

$$\frac{\sum_{j=1}^{i+1} |x_j|}{|f(x)|} \leq \frac{i+1}{n-1}$$

quindi

$$e_{alg} = \frac{1}{|f(x)|} \sum_{i=1}^{n-1} |\sum_{j=1}^{i+1} x_j| * |\varepsilon_i|$$

$$\leq u \frac{1}{|f(x)|} \sum_{i=1}^{n-1} |\sum_{j=1}^{i+1} x_j|$$

$$\leq \frac{u}{n-1} \sum_{i=1}^{n-1} (i+1)$$

Esercizio 1.6

Verificare che

$$\sum_{i=1}^{n-1} (i+1) = (n^2 + n - 2)/2$$

Verifichiamo per induzione: per $n = 2$, si ha:

$$\sum_{i=1}^{2-1} (i+1) = 2 = (2 * 2 + 2 - 2)/2 = 2$$

Supponiamo vero l'asserto per $n-1$, e lo dimostriamo per n.

$$\sum_{i=1}^{n-1}(i+1) =$$

Isolando l'ultimo termine della somma:

$$n + \sum_{i=1}^{(n-1)-1}(i+1) =$$

e applicando il passo induttivo al secondo addendo si ottiene:

$$n + [(n-1)^2 + (n-1) - 2]/2 =$$

$$n + [n^2 + 1 - 2n + n - 1 - 2]/2 =$$

$$n + [n^2 - n - 2]/2 =$$

$$[n^2 + 2n - n - 2]/2 = \frac{n^2 + n - 2}{2}$$

Applicando l'esercizio e sostituendo nella formula 1 si ottiene:

$$e_{alg} < \frac{u}{n-1}\sum_{i=1}^{n-1}(i+1)$$

$$< (n^2 + n - 2)/2 * \frac{u}{n-1} < (n+1)/2u$$

e l'errore è dell'ordine di $u/2$.

Procedimento 2: supponiamo di avere $n = 2^3$ numeri da sommare. Allora

$$x_1 + x_2 = z_1,\ x_3 + x_4 = z_2,\ x_5 + x_6 = z_3,\ x_7 + x_8 = x_4$$

$$z_1 + z_2 = z_5,\ z_3 + z_4 = z_6$$

$$z_5 + z_6 = z_7$$

$$|E_r| > \log n$$

1.6.6 Errore nel calcolo dell'esponenziale

Calcoliamo

$$f(x, y) = e^{x+y} = e^x * e^y$$

In questo caso c'è una funzione di libreria $\exp(z)$ tale che per ogni z, $\exp(z) = e^z * (1 + \varepsilon_z)$, dove $\varepsilon = |e_{an} + e_{alg}| \leq u$.

1. Calcoliamo l'errore inerente:

$$c_x = \frac{x}{f(x,y)} * \frac{\partial f}{\partial x} = \frac{x}{e^{x+y}} * e^{x+y} = x$$

$$c_y = y$$

quindi

$$e_{in} = x * \varepsilon_x + y * \varepsilon_y$$

2. Applicando l'algoritmo $A_1 = e^x * e^y$ si ha:

$$z_1 = e^x,\ z_2 = e^y,\ z_3 = z_1 * z_2$$

Rappresentiamo il grafo:

$$\begin{array}{ccccc} x & & & & y \\ \uparrow (c_x) & & & & \uparrow (c_y) \\ & z_1 & & z_2 & \\ & \uparrow (c_1) & & \uparrow (c_2) & \\ & & z_3 & & \end{array}$$

Sapendo che $c_1 = c_2 = 1$ si ha:

$$e_{alg} = \varepsilon_3 + 1 * \varepsilon_1 + 1 * \varepsilon_2 \leq |3u|$$

Questo è un algoritmo stabile qualsiasi siano i due numeri x e y.

3. Per via analitica, supponendo che $\mathrm{fl}(x) = x$ e $\mathrm{fl}(y) = y$ calcoliamo l'errore del primo algoritmo:

$$e^{x+y} = e^x * e^y = \mathrm{fl}[\mathrm{fl}(e^x) * \mathrm{fl}(e^y)]$$

$$= \mathrm{fl}[e^x * (1 + \varepsilon_1) * e^y * (1 + \varepsilon_2)]$$

$$= (1 + \varepsilon_3) * e^x * (1 + \varepsilon_1) * e^y * (1 + \varepsilon_2)$$

e al primo ordine:

$$= e^x * e^y * (1 + \varepsilon_1 + \varepsilon_2 + \varepsilon_3)$$

quindi

$$E_r = \frac{\mathrm{fl}(e^{x+y}) - e^{x+y}}{e^{x+y}}$$

$$= \frac{e^x * e^y * (1 + \varepsilon_1 + \varepsilon_2 + \varepsilon_3) - e^x * e^y}{e^x * e^y}$$

$$= |\varepsilon_1 + \varepsilon_2 + \varepsilon_3| \leq |\varepsilon_1| + |\varepsilon_2| + |\varepsilon_3| < 3u$$

4. Applichiamo l'algoritmo 2 $f(x, y) = e^{x+y}$ e poniamo

$$w_1 = x + y,\ w_2 = e^{w_1}$$

$$\begin{array}{ccc} x & & y \\ \uparrow (c_x) & & \uparrow (c_y) \\ & w_1 & \\ & \uparrow (c_1) & \\ & w_2 & \end{array}$$

e assumendo $\varepsilon_x = \varepsilon_y = 0$, e sapendo che $c_1 = w_1$ si ha:

$$|e_{alg}| = \varepsilon_2 + w_1\varepsilon_1 = \varepsilon_2 + (x+y)\varepsilon_1 < (1+x+y)u$$

Quando $x+y$ è piccolo, l'algoritmo 2 è più stabile, mentre è meno stabile quando $x+y$ è grande.

5. Per via analitica, calcolo l'errore algoritmico del secondo algoritmo.

$$e^{x+y} = \mathrm{fl}[e^{\mathrm{fl}(x+y)}]$$

$$= \mathrm{fl}(e^{(1+\varepsilon_1)*(x+y)}$$

$$= e^{(1+\varepsilon_1)*(x+y)} * (1+\varepsilon_2)$$

Al primo ordine:

$$= [e^{x+y} + e^{x+y} * \varepsilon_1] * (1+\varepsilon_2)$$

$$= e^{x+y} + (x+y) * e^{x+y} * \varepsilon_1 + e^{x+y} * \varepsilon_2$$

Calcolo l'errore relativo:

$$E_r = \frac{e^{x+y} + e^{x+y} * (x+y) * \varepsilon_1 + e^{x+y} * \varepsilon_2 - e^{x+y}}{e^{x+y}}$$

e al primo ordine rimane:

$$= (x+y)\varepsilon_1 + \varepsilon_2$$

1.7 Comportamento dell'errore algoritmico e dell'errore analitico

Data una funzione f non razionale, l'errore totale al primo ordine è $E_{alg} + E_{an} + E_{in}$.

Se abbiamo una funzione non razionale, si ha $E_{an} \neq 0$. Ci si aspetta che per n crescente, l'errore analitico si riduca. Invece, per n crescente, E_{alg} aumenta.

Grafico: supponiamo di avere un grafico con n in ascissa e l'errore in ordinata. L'errore analitico è rappresentato da una curva che decresce, viceversa l'errore algoritmico è una curva crescente. Osservo che abbiamo un valore n_0 per cui abbiamo la situazione migliore possibile, cioè il polinomio è un'approssimazione sufficientemente buona della funzione, e l'errore algoritmico non è ancora troppo elevato (punto di bilanciamento). La curva complessiva dell'errore relativo (somma di errore algoritmico e analitico) ha una forma di V, raggiunge una cuspide (n_0) e poi risale. Infatti inizialmente, si aumenta n per ridurre l'errore analitico, ma l'aumentare di n provoca l'aumentare dell'errore algoritmico, e quindi non si riesce a ottenere una riduzione soddisfacente dell'errore.

1.7.1 Esempio 1.8

Consideriamo l'esempio di approssimazione della derivata prima con il rapporto incrementale.

$$f'(x) = \frac{f(x+h) - f(x)}{h} + o(h)$$

o con il rapporto incrementale centrato:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + o(h^2)$$

Grafico: rappresentiamo h sull'asse delle ascisse, e in ordinata l'errore relativo (trascuriamo l'errore inerente). Al decrescere di h, l'errore algoritmico aumenta, e i risultati guadagnati riducendo h non portano vantaggi. Considerando invece il rapporto incrementale centrato, si ha un miglioramento, ma anche in questo caso, al diminuire di h, l'errore analitico diminuisce ma a partire da un certo punto l'errore algoritmico aumenta e deteriora il risultato.

Conclusione importante: su un calcolatore, non vale la regola che più fine è l'approssimazione, più l'errore si riduce.

2

Sistemi lineari

2.1 Norme di vettori e di matrici

2.1.1 Norma di vettori

Nella risoluzione di sistemi lineari si ha l'obiettivo di misurare quantità non scalari.

◊ **Definizione** Una **norma vettoriale** è una funzione da $\mathbb{C}^n$ $\mathbb{R}^+ \setminus 0$ tale che

1. $|\boldsymbol{x}| \geq 0$ e $|\boldsymbol{x}| = 0 \iff \boldsymbol{x} = 0$
2. $|\alpha \boldsymbol{x}| = |\alpha| * |\boldsymbol{x}|$ (omogeneità)
3. $|\boldsymbol{x} + \boldsymbol{y}| \leq |\boldsymbol{x}| + |\boldsymbol{y}|$ (disuguaglianza triangolare)

Esempio 2.1

Esempi di norme sono:

1. $|\boldsymbol{x}| = \sqrt{\sum_{i=1}^{n} x_i^2}$ norma 2
2. $|\boldsymbol{x}| = \sum_{i=1}^{n} |x_i|$ norma 1
3. $|\boldsymbol{x}|^\infty = \max i |x_i|$ norma infinito

e gli intorni in queste tre norme sono rispettivamente un cerchio, un rombo e un quadrato.

Proprietà 1: la norma è una funzione uniformemente continua, cioè

$$\forall \varepsilon > 0 \, \exists \delta \, t.c. |\boldsymbol{x} - \boldsymbol{y}| < \delta \longrightarrow |\boldsymbol{x}| - |\boldsymbol{y}| \leq \varepsilon$$

A vettori vicini corrispondono norme vicine indipendentemente dagli ordini di grandezza.

Proprietà 2: vale la proprietà di equivalenza topologica delle norme. Supponiamo di prendere due norme qualsiasi $|\boldsymbol{x}|_1$ e $|\boldsymbol{x}|_2$, con n fissato. Allora esistono $\alpha, \beta \in \mathbb{R}$ con $0 < \alpha < \beta$ tali che

$$\alpha |\boldsymbol{x}|_2 \leq |\boldsymbol{x}|_1 \leq \beta |\boldsymbol{x}|_2$$

Allora, siccome tutte le norme sono equivalenti, si può scegliere la norma che dà meno errori.

Proprietà 3: per ogni $\boldsymbol{x} \in \mathbb{C}^n$, considerate le norme 1,2 e ∞ dell'esempio precedente, si ha:

$$|\boldsymbol{x}|^\infty \leq |\boldsymbol{x}|_2 \leq \sqrt{n}|\boldsymbol{x}|^\infty$$

$$|\boldsymbol{x}|_2 \leq |\boldsymbol{x}|_1 \leq \sqrt{n}|\boldsymbol{x}|_2$$

e, unendo le due relazioni:

$$|x|^\infty \leq |\boldsymbol{x}|_2 \leq |\boldsymbol{x}|_1$$

2.1.2 Norma matriciale

◊ **Definizione** Una **norma matriciale** è una funzione : $\mathbb{C}^{n^2} \to \mathbb{R}^+ \setminus 0$ tale che

1. $|X| = 0$ solo se X è la matrice nulla
2. $|\alpha X| = |\alpha||X|$
3. $|X + Y| \leq |X| + |Y|$
4. $|XY| \leq |X| * |Y|$

Proprietà 1,2: valgono l'uniforme continuità e l'equivalenza delle norme come conseguenze dei punti 1,2,3.

Esempio 2.2

È necessario imporre anche la quarta proprietà. Infatti, data la matrice

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}$$

$$B = \begin{pmatrix} 1 & 6 \\ 1 & 8 \\ 3 & 4 \end{pmatrix}$$

si ha che AB è la matrice nulla, e quindi vale

$$0 < |A| * |B|$$

2.1.3 Relazioni tra norme vettoriali e norme matriciali

◊ **Definizione** La norma $|\dots|_*$ è una **norma matriciale compatibile con una certa norma vettoriale** $|\dots|_v$ se

$$|A\boldsymbol{x}|_v \leq |A|_* * |\boldsymbol{x}|_v, \forall \boldsymbol{x} \in \mathbb{C}^n, \forall A \in \mathbb{C}^{n^2}$$

◊ **Definizione** Definiamo la **norma matriciale indotta da una norma vettoriale fissata** come

$$|A| = \max_{\boldsymbol{x}\in\mathbb{C}^n \, t.c. \, |\boldsymbol{x}|_v=1} |A\boldsymbol{x}|_v$$

equivalentemente

$$|A| = \sup_{\boldsymbol{x}\in\mathbb{C}^n} |\frac{A\boldsymbol{x}}{|A\boldsymbol{x}|_v}|_v$$

Proprietà 3: Una norma vettoriale indotta è una norma matriciale compatibile.

Dim. Se $\boldsymbol{x} = 0$, la definizione di norma matriciale compatibile è verificata. Se $\boldsymbol{x} \neq 0$, si ha

$$|A\boldsymbol{x}|_v = \frac{|A\boldsymbol{x}|_v}{|\boldsymbol{x}|_v} * |\boldsymbol{x}|_v$$

$$= |\boldsymbol{x}|_v * |A\frac{\boldsymbol{x}}{|\boldsymbol{x}|_v}|_v =$$

$$= |\boldsymbol{x}|_v * |A\boldsymbol{y}|_v$$

$$\leq |\boldsymbol{x}|_v * \max_{\boldsymbol{y}\, t.c. |\boldsymbol{y}|_v=1} |A\boldsymbol{y}|_v$$

$$\leq |\boldsymbol{x}|_v * |A|$$

cvd

Proprietà 4: La norma matriciale indotta è la più piccola tra le norme matriciali compatibili con la norma vettoriale considerata.

Dim.

$$|A| = \max_{\boldsymbol{x}\, t.c. \, |\boldsymbol{x}|=1} |A\boldsymbol{x}|_v$$

e per la compatibilità della norma $*$:

$$\leq \max_{\boldsymbol{x}\, t.c. \, |\boldsymbol{x}|_v=1} |A|^* |\boldsymbol{x}|_v$$

$$\leq |A|^*$$

cvd

2.1.4 Norme matriciali indotte dalle principali norme vettoriali

Proprietà 5:

norma vettoriale	norma matriciale compatibile
norma 1	$\max_{j=1}^n [\sum_{i=1}^n \lvert A_{ij}\rvert]$
norma infinito	$\max_{i=1}^n [\sum_{j=1}^n \lvert A_{ij}\rvert]$
norma 2	$\rho^{1/2}(A_h * A)$

Nell'espressione della norma matriciale indotta dalla norma 2, cioè dalla norma euclidea, ρ rappresenta il raggio spettrale.

◊ **Definizione** Consideriamo $B \in \mathrm{Mat}_{n,n}$, allora

$$\rho = \max_i |\lambda_i|,$$

cioè il raggio spettrale è il massimo dei moduli degli autovalori della matrice.

◊ **Definizione** A_h è la **trasposta coniugata**, cioè se A ha elementi reali, si ha $(A_h)_{ij} = A_{ji}$, mentre se $A \in \mathcal{C}$, $(A_h)_{ij} = \bar{A}_{ji}$.

Proprietà 6: Per ogni $A \in \mathbb{C}^n$ valgono le seguenti disuguaglianze:

$$\begin{aligned}
\frac{1}{\sqrt{n}}|A|^\infty \leq |A|_2 &\qquad \leq \sqrt{n}|A|^\infty \\
\frac{1}{\sqrt{n}}|A|_1 \leq |A|_2 &\qquad \leq \sqrt{n}|A|_1 \\
\max_{i,j}|A_{ij}| \leq |A|_2 &\qquad \leq \max_{i,j}|A_{ij}|??? \\
|A|_2 \leq \sqrt{n}|A|_2 &\qquad \leq |A|^\infty
\end{aligned}$$

Osservazione 2.1

Se $A = A_h$, allora $|A|^\infty = |A|_1$, e

$$|A|_2 = \rho^{1/2}(A^2).$$

ma siccome per gli autovalori vale la relazione $A\boldsymbol{x} = \lambda\boldsymbol{x}$, moltiplicando per A ottengo:

$$A^2\boldsymbol{x} = \lambda A\boldsymbol{x} = \lambda^2\boldsymbol{x}$$

cioè gli autovalori di A^2 sono i quadrati degli autovalori di A.

Quindi

$$|A|_2 = \rho(A)$$

Se A è hermitiana definita positiva, cioè $\forall \boldsymbol{x} \neq 0$, $\boldsymbol{x} \cdot A\boldsymbol{x} > 0$, allora gli autovalori sono positivi, e si ha

$$|A|_2 = \max_i \lambda_i(A)$$

Nota: il raggio spettrale non è una norma matriciale, a meno che non mi limiti a sottoclassi di matrici.

Esistono infatti A, B tali che

$$\rho(A + B) > \rho(A) + \rho(B)$$

Esempio 2.3

Considerando

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

$$B = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$$

Allora si verifica che

$$\rho(A) = \rho(B) = 0$$

mentre

$$\rho(A + B) = 1$$

e la disuguaglianza triangolare è violata.

2.1.5 Altre proprietà della norma matriciale indotta

Proprietà 7: per ogni norma matriciale indotta, vale che $\rho(A) \leq |A|^*$.

Dim. Consideriamo la relazione

$$A\boldsymbol{x} = \lambda\boldsymbol{x}$$

$$|\lambda\boldsymbol{x}| = |A\boldsymbol{x}|_v$$

$$\longrightarrow |\lambda||\boldsymbol{x}|_v = |A\boldsymbol{x}|_v$$

$$\longrightarrow |\lambda| = |A\frac{\boldsymbol{x}}{|\boldsymbol{x}|_v}|_v$$

$$|\lambda| \leq \max_{\boldsymbol{v}\, t.c.\, |\boldsymbol{v}|=1} |A\boldsymbol{v}|_v = |A|^*$$

cioè

$$|\lambda| \leq |A|^*$$

vale per ogni autovalore, e quindi anche per quello che realizza il raggio spettrale, cioè

$$\rho(A) \leq |A|^*$$

cvd

2.1.6 Numero di condizionamento

◊ **Definizione** Si dice numero di condizionamento di una matrice rispetto ad una fissata norma la quantità

$$K(A) = |A| * |A^{-1}|$$

Consideriamo una funzione $g\colon \mathbb{R}^n \to \mathbb{R}^n$, tale che $Ag(P) = P$.

$$\bar{P} = P + \delta_P, \longrightarrow g(\bar{P}) \simeq g(P) + J_g(P) * \delta_P$$

Valutando l'errore relativo

$$|E_r| = \frac{|g(\bar{P}) - g(P)|}{|g(P)|}$$

$$= \frac{|g(P) + J_g(P) * \delta_P - g(P)|}{|g(P)|}$$

$$= \frac{|J_g(P)\delta_P|}{|g(P)|}$$

$$\leq \frac{|J_g(P)| * |\delta_P|}{|G(P)|}$$

Moltiplichiamo e dividiamo per $|P|$.

$$E_r \leq |J_g(P)| * \frac{|P|}{|g(P)|} * \frac{|\delta_P|}{|P|}$$

Siccome stiamo supponendo di risolvere il sistema

$$Ag(P) = P$$

segue che

$$J_g(P) = A^{-1}$$

$$g(P) = A^{-1}P$$

quindi

$$E_r \leq |A^{-1}| * \frac{|Ag(P)|}{|g(P)|} * \varepsilon_P$$

tenendo conto che la norma matriciale utilizzata è compatibile:

$$E_r \leq |A| * \frac{|A^{-1}| * |G(P)|}{|g(P)|} * \frac{|\delta_P|}{|P|}$$

e ponendo $\varepsilon_x = \frac{|\delta_P|}{|P|}$ si ottiene

$$E_r \leq K(A) * \varepsilon_X$$

2.1.7 Proprietà del numero di condizionamento

◊ **Definizione** Il numero di condizionamento di una matrice A, supposta non singolare, è $K(A) = |A| * |A^{-1}|$. Si dice che $K(A)$ è infinito se A è singolare.

Lemma 2.1: Per ogni norma, $K(A) \geq 1$.

Dim.

$$K(A) = |A| * |A^{-1}| \geq |A^{-1}A| = |I| \geq 1$$

cvd

2.1.8 Condizionamento spettrale

◊ **Definizione** Si definisce condizionamento spettrale la quantità $K_2(A) = |A|_2 * |A^{-1}|_2$ (infatti $|A|_2 = \rho^{1/2}(A_hA)$ con ρ raggio spettrale).

A_HA è definita positiva, se A è non singolare. Infatti $\boldsymbol{x} \cdot (A_hA\boldsymbol{x}) = \boldsymbol{x} \cdot (A_H\boldsymbol{y}) \geq 0 \forall \boldsymbol{x} \neq 0, \quad \boldsymbol{y} = A\boldsymbol{x} \neq 0$.

Se A è definita positiva, allora ha tutti autovalori reali positivi, e

$$\rho^{1/2}(A_hA) = \sqrt{\max_i \lambda_i(A_hA)} = \sqrt{\lambda_{max}(A_hA)}$$

dove i λ_i sono autovalori di A_hA.

Inoltre

$$|A^{-1}|_2 = \rho^{1/2}(A_h^{-1}A^{-1}) = \rho^{1/2}(AA_h)^{-1} = \sqrt{\max \lambda_i((AA_h)^{-1}}, \quad \lambda_i \text{autovalori di} (AA_h)^{-1}$$

Allora

$$|A^{-1}|_2 = \max_i [\frac{1}{\lambda_i(AA_H)}] = \frac{1}{\lambda_{min}(AA_h)}$$

e siccome A h AA_h e A_hA sono simili e hanno quindi gli stessi autovalori, allora

$$|A^{-1}|_2 = \frac{1}{\lambda_{min}(A_hA)}.$$

Allora, se A è non singolare,

$$K_2(A) = \sqrt{\frac{\lambda_{max}(A_hA)}{\lambda_{min}(A_hA)}}.$$

Se $A_h = A$, allora

$$K_2(A) = \frac{\max|\lambda(A)|}{\min|\lambda(A)|}$$

e se A è anche definita positiva

$$K_2(A) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$$

2.1.9 Stime del numero di condizionamento

Consideriamo una norma matriciale indotta. Vale la seguente proprietà: $\rho^{1/2}(A) \leq |A|$, allora

$$K(A) = |A| * |A^{-1}| \geq \rho(A) * \rho(A^{-1}) = \frac{\max |\lambda(A)|}{\min |\lambda(A)|}$$

Determiniamo le matrici con numeri di condizionamento molto alti o molto bassi.

Esempio 2.4 [matrice di Hilbert]

La matrice di Hilbert ha entrate della forma

$$H_{ij} = \frac{1}{i+j-1}$$

Ad esempio, in dimensione 3

$$H_3 = \begin{matrix} 1 & 1/2 & 1/3 \\ 1/2 & 1/3 & 1/4 \\ 1/3 & 1/4 & 1/5 \end{matrix}$$

Nella tabella rappresento l'aumentare del numero di condizionamento al crescere di n:

valore di n	condizionamento
2	10^0
3	10^3
5	10^5
6	10^7
10	10^{13}

◊ **Definizione** Se $K_2(A) = 1$ si dice che la matrice è perfettamente condizionata.

Esempio 2.5

Tra le matrici per cui $K(A) = 1$ c'è la matrice identica.

$$K_2(id) = 1$$

Anche le matrici tali che

$$A_h A = A A_h = id$$

cioè tali che

$$A^{-1} = A_h$$

sono perfettamente condizionate, infatti:

$$|A|_2 = 1$$

$$|A^{-1}|_2 = 1$$

Osservazione 2.2

Vale la proprietà

$$\det \alpha A = \alpha^n \det A, \quad A \in \mathrm{Mat}_n$$

$$\det[0.1 * A] = (0.1)^n A \to 0$$

ma la matrice non è singolare.

Il determinante non è una buona misura della singolarità della matrice.

2.2 Sistemi lineari e errore inerente (condizionamento)

2.2.1 Perturbazioni del termine noto

Consideriamo un sistema della forma

$$A\boldsymbol{x} = \boldsymbol{b}, \quad A \in \mathrm{Mat}_{n,\mathbb{R}}$$

e supponiamo che A sia non singolare.

Consideriamo il caso in cui

$$A\boldsymbol{x} = \bar{\boldsymbol{b}}$$

$$\bar{\boldsymbol{b}} = \boldsymbol{b} + \delta_b$$

che ha soluzione $\bar{\boldsymbol{x}} = \boldsymbol{x} + \delta_x$. Ci chiediamo quale sia la relazione tra δ_x e δ_b.

Considerando le relazioni

$$A\boldsymbol{x} = \boldsymbol{b}$$

$$A\bar{\boldsymbol{x}} = \bar{\boldsymbol{b}}$$

Sottraiamo membro a membro:

$$A\bar{\boldsymbol{x}} - A\boldsymbol{x} = \bar{\boldsymbol{b}} - \boldsymbol{b}$$

$$\bar{\boldsymbol{x}} - \boldsymbol{x} = A^{-1}[\bar{\boldsymbol{b}} - \boldsymbol{b}] = A^{-1}\delta_b$$

passando alle norme

$$|\bar{\boldsymbol{x}} - \boldsymbol{x}| = |A^{-1}\delta_b| \leq |A^{-1}| * |\delta_b|$$

$$|\bar{\boldsymbol{x}} - \boldsymbol{x}| \leq |A^{-1}| * |\delta_b| \quad \text{relazione}\circ$$

ma

$$|\boldsymbol{b}| \leq |A| * |\boldsymbol{x}|$$

allora

$$\frac{1}{|\boldsymbol{x}|} \leq \frac{|A|}{|\boldsymbol{b}|} \quad \text{relazione} \circ \circ$$

e unendo le relazioni $\circ$ e $\circ\circ$:

$$\frac{|\bar{\boldsymbol{x}} - \boldsymbol{x}|}{|\boldsymbol{x}|} \leq \frac{|A| * |A^{-1}| * |\delta_b|}{|\boldsymbol{b}|}$$

quindi

$$\frac{|\bar{\boldsymbol{x}} - \boldsymbol{x}|}{|\boldsymbol{x}|} \leq K(A) * \frac{|\delta_b|}{|\boldsymbol{b}|}$$

e in termini degli errori:

$$\varepsilon_x \leq K(A)\varepsilon_b$$

2.2.2 Caso generale

Consideriamo un sistema in cui ci sia una perturbazione anche sui coefficienti della matrice. Risolviamo quindi l'equazione

$$\bar{A}\bar{\boldsymbol{x}} = \bar{\boldsymbol{b}}, \quad \bar{A} = A + \delta_A, \quad \bar{\boldsymbol{b}} = \boldsymbol{b} + \delta_b$$

Sono necessari i due seguenti teoremi tecnici.

Teorema 2.1

Data una norma matriciale indotta, tale che $|A| < 1$ allora $I + A$ è non singolare, e $|(I + A)^{-1}| \leq \frac{1}{1-|A|}$.

Dimostrazione

Dimostriamo che $I + A$ è non singolare.

$$|A| < 1 \longrightarrow \rho(A) < 1,$$

gli autovalori di $I + A$ sono della forma $1 + \lambda_i$ con λ_i autovalori di A.

Consideriamo la relazione

$$(I + A)(I + A)^{-1} = I$$

ed espandendo il prodotto:

$$(I + A)^{-1} + A(I + A)^{-1} = I$$

$$(I + A)^{-1} = I - A(I + A)^{-1}$$

$$|(I + A)^{-1}| \leq |I - A(I + A)^{-1}| \leq |I| + |A| * |(I + A)^{-1}|$$

$|I| = 1$ perché sto considerando una norma matriciale indotta.

$$(1 - |A|) * |(I + A)^{-1}| \leq 1$$

$1 - |A| > 0$, allora

$$|(I + A)^{-1}| \leq \frac{1}{1 - |A|}$$

cvd

Teorema 2.2

Se $|A^{-1}| * |\delta_A| < 1$ allora $\bar{A}$ è non singolare, e

$$|(I + A^{-1}\delta_A)^{-1}| \leq \frac{1}{1 - |A^{-1}| * |\delta_A|}$$

Dimostrazione

$\bar{A} = A + \delta_A$, e siccome A è non singolare posso raccogliere A e scrivere:

$$\bar{A} = A(I + A^{-1}\delta_A)$$

ma $|A^{-1}\delta_A| \leq |A^{-1}| * |\delta_A| < 1$ per ipotesi. Allora $\bar{A}$ è non singolare, per il teorema precedente.

$$|(I + A^{-1}\delta_A)^{-1}| \leq \frac{1}{1 - |A^{-1}\delta_A|} \leq \frac{1}{1 - |A^{-1}||\delta_A|}$$

cvd

2.2.3 Risultato di analisi a priori

Teorema 2.3

Dato il sistema $A\boldsymbol{x} = \boldsymbol{b}$, con A non singolare, $\boldsymbol{b} \neq 0$, con le perturbazioni sui dati δ_A, δ_b, allora consideriamo il sistema

$$(A + \delta_A)(\boldsymbol{x} + \delta_x) = \boldsymbol{b} + \delta_b$$

nell'ipotesi che

$$|A^{-1}| * |\delta_A| < 1$$

allora

$$\bar{A} = A + \delta_A$$

è non singolare e per l'errore relativo

$$\varepsilon_x = \frac{|\delta_x|}{|\boldsymbol{x}|} = \frac{|\bar{\boldsymbol{x}} - \boldsymbol{x}|}{|\boldsymbol{x}|}$$

vale la relazione

$$\varepsilon_x \leq \frac{K(A) * (\varepsilon_b + \varepsilon_A)}{1 - K(A) * \varepsilon_A}$$

con

$$\varepsilon_A = \frac{|\delta_A|}{|A|}$$

Il teorema afferma che se il numero di condizionamento di A è piccolo, a errori piccoli sui dati corrisponde un errore piccolo nella soluzione, e il problema è ben condizionato, se invece $K(A)$ è grande piccoli errori sui dati possono dare grandi errori sulla soluzione (in quest'ultimo caso, la quantità al secondo membro è molto grande e, su quella a primo membro, che deve essere minore, non si hanno molte informazioni).

Dimostrazione

$$(A + \delta_A)(\boldsymbol{x} + \delta_x) = \boldsymbol{b} + \delta_b$$

$$A\boldsymbol{x} + \boldsymbol{x}\delta_A + A\delta_x + \delta_A\delta_x = \boldsymbol{b} + \delta_b$$

e siccome $A\boldsymbol{x} = \boldsymbol{b}$,

$$\boldsymbol{x}\delta_A + A\delta_x + \delta_A\delta_x = \delta_b$$

e isolando $A\delta_x$:

$$A\delta_x = \delta_b - \boldsymbol{x}\delta_A - \delta_A\delta_x$$

e siccome A è non singolare

$$\delta_x = A^{-1}[\delta_b - \boldsymbol{x}\delta_A - \delta_A\delta_x]$$

e portando a primo membro i termini moltiplicati per δ_x:

$$(I + A^{-1}\delta_A)\delta_x = A^{-1}[\delta_b - \boldsymbol{x}\delta_A]$$

e equivalentemente

$$\delta_x = (I + A^{-1}\delta_A)^{-1} * A^{-1}[\delta_b - \boldsymbol{x}\delta_A]$$

Passando alla norma:

$$|\delta_x| \leq |(I + A^{-1}\delta_A)^{-1}| * |A^{-1}[\delta_b - \boldsymbol{x}\delta_A]|$$

e per i due teoremi dimostrati prima possiamo maggiorare $|(I + A^{-1}\delta_A)^{-1}|$:

$$|\delta_x| \leq \frac{1}{1 - |A^{-1}| * |\delta_A|} * |A^{-1}[\delta_b - \boldsymbol{x}\delta_A]|$$

$$(1 - |A^{-1}| * |\delta_A|)|\delta_x| \leq |A^{-1}| * [|\delta_b| + |\boldsymbol{x}| * |\delta_A|]$$

Dividiamo entrambi i membri per $|\boldsymbol{x}|$:

$$(1 - |A^{-1}| * |\delta_A|) * \frac{|\delta_x|}{|\boldsymbol{x}|} \leq \frac{|A^{-1}| * [|\delta_b| + |\boldsymbol{x}| * |\delta_A|]}{|\boldsymbol{x}|}$$

$$(1 - |A^{-1}| * |\delta_A|) * \frac{|\delta_x|}{|\boldsymbol{x}|} \leq |A^{-1}| * [\frac{|\delta_b|}{|\boldsymbol{x}|} + |\delta_A|]$$

ponendo $\varepsilon_x = \frac{|\delta_x|}{|\boldsymbol{x}|}$ e sapendo che

$$\frac{1}{|\boldsymbol{x}|} \leq \frac{|A|}{|\boldsymbol{b}|}$$

allora

$$(1 - |A^{-1}| * |\delta_A|) * \varepsilon_x \leq |A^{-1}| * [\frac{|A||\delta_b|}{|\boldsymbol{b}|} + |\delta_A|]$$

Moltiplicando e dividendo l'ultimo addendo per $|A|$ si ottiene:

$$(1 - |A^{-1}| * |\delta_A|) * \varepsilon_x \leq |A^{-1}| * |A|[\frac{|\delta_b|}{|\boldsymbol{b}|} + \frac{|\delta_A|}{|A|}]$$

$$(1 - |A^{-1}| * |\delta_A|) * \varepsilon_x \leq K(A) * (\varepsilon_A + \varepsilon_b)$$

infine, sapendo che

$$|A^{-1}| * |\delta_A| = \frac{|\delta_A|}{|A|} * K(A) = K(A) * \varepsilon_A$$

si ha

$$\varepsilon_x \leq \frac{K(A) * (\varepsilon_A + \varepsilon_b)}{1 - K(A)\varepsilon_A}$$

cvd

2.2.4 Risultato di analisi a posteriori

Teorema 2.4

Sia $A\boldsymbol{x} = \boldsymbol{b}$, A non singolare, $\boldsymbol{b} \neq 0$, e $\bar{\boldsymbol{x}}$ soluzione calcolata in aritmetica *floating point*. Allora l'errore relativo su $\boldsymbol{x}$ è

$$\varepsilon_x \leq K(A)\frac{|r|}{|\boldsymbol{b}|}$$

dove $r = r(\bar{\boldsymbol{x}}) = \boldsymbol{b} - A\bar{\boldsymbol{x}}$ (residuo).

Si considera il sistema lineare come un'equazione vettoriale di cui cerchiamo la radice. Se chiamiamo x_* la soluzione esatta, $r(x_*) = 0$.

Se il residuo è piccolo ma il $K(A)$ è grande, allora la maggiorazione non dà informazioni importanti.

Dimostrazione

$$r = \boldsymbol{b} - A\bar{\boldsymbol{x}}$$

$$\boldsymbol{b} = A\boldsymbol{x} \longrightarrow r = A\boldsymbol{x} - A\bar{\boldsymbol{x}} = A(\boldsymbol{x} - \bar{\boldsymbol{x}})$$

ovvero $\boldsymbol{x} - \bar{\boldsymbol{x}} = A^{-1}r$.

Passando alla norma

$$|\boldsymbol{x} - \bar{\boldsymbol{x}}| \leq |A^{-1}||r|$$

$$|\boldsymbol{x} - \bar{\boldsymbol{x}}| \leq K(A) * \frac{|r|}{|A|}$$

e siccome $\boldsymbol{b} = A\boldsymbol{x}$

$$|\boldsymbol{b}| \leq |A||\boldsymbol{x}|$$

$$\frac{1}{|A|} \leq \frac{|\boldsymbol{x}|}{|\boldsymbol{b}|}$$

sostituendo

$$|\boldsymbol{x} - \bar{\boldsymbol{x}}| \leq K(A)\frac{|r||\boldsymbol{x}|}{|\boldsymbol{b}|}$$

e dividendo ambo i membri della disuguaglianza per $|\boldsymbol{x}|$

$$E_r \leq K(A)\frac{|r|}{|\boldsymbol{b}|}$$

cvd

2.3 Sistemi lineari su un calcolatore

2.3.1 Casi semplici

Vogliamo risolvere un sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ con A matrice quadrata di dimensione n supposta non singolare, e $\boldsymbol{x}, \boldsymbol{b} \in \mathbb{R}^n$.

1. Se A è una matrice diagonale, ci sono equazioni disaccoppiate, e, per i coefficienti della soluzione, vale la formula:

$$x_i = \frac{b_i}{a_i}$$

2. Nel caso di una matrice triangolare inferiore $A = L$, nell'equazione i-esima esplicito l'incognita x_i:

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} a_j x_j}{a_{ii}} \forall i = 1, \ldots, n$$

infatti risolvendo l'i-esima equazione, conosciamo già tutte le $i-1$ incognite precedenti, ricavate nelle equazioni sopra esplicitando le incognite precedenti (risoluzione *forward*).

3. Nel caso in cui $A = U$ triangolare superiore, si applica lo stesso procedimento ma partendo dall'ultima incognita.

$$x_i = \frac{(b_i - \sum_{j=i+1}^{n} a_j x_j)}{a_{ii}}, \quad i = n : 1$$

(risoluzione *backward*)

Nota: le formule risolutive, e la divisione per a_i è ben posta perché siccome $\det A \neq 0$, i coefficienti sulla diagonale di una matrice triangolare sono diversi da 0.

I metodi per la risoluzione di sistemi lineari si distinguono in

1. metodi diretti (di fattorizzazione)
2. metodi iterativi, usati per matrici di grosse dimensioni.

2.3.2 Metodi di fattorizzazione

Una matrice A può essere scritta come prodotto BC, con B, C facilmente invertibili, in modo che i sistemi

$$C\boldsymbol{x} = \boldsymbol{y}$$

$$B\boldsymbol{y} = \boldsymbol{b}$$

siano facilmente risolubili.

Siccome $A = BC$, sostituendo nel sistema originario ottengo

$$BC\boldsymbol{x} = \boldsymbol{b}$$

e ponendo $C\boldsymbol{x} = \boldsymbol{y}$ ottengo

$$B\boldsymbol{y} = \boldsymbol{b}$$

e lo risolvo. Una volta ottenuto $\boldsymbol{y}$, per trovare $\boldsymbol{x}$ risolvo quindi

$$C\boldsymbol{x} = \boldsymbol{y}$$

2.3.3 Fattorizzazioni classiche

Esistono tre possibili fattorizzazioni della matrice A:

1. fattorizzazione $PA = LU$, con la richiesta che la diagonale principale di L sia composta da 1, e dove P è una matrice di permutazione.
2. se A è simmetrica definita positiva, si può considerare la fattorizzazione $A = LL^H = U^hU$, con L^h coniugata trasposta di L, con la richiesta che gli elementi della diagonale principale siano positivi.
3. $A = QR$, dove Q è una matrice unitaria ($QQ^h = Q^hQ = 1$) e R è una triangolare superiore.

2.4 Fattorizzazione LU (eliminazione gaussiana)

2.4.1 Eliminazione gaussiana

Consideriamo la matrice A e facciamo operazioni sulle sue righe.

$$A\boldsymbol{x} = [\boldsymbol{b}]^i, \quad i = 1, \dots, n$$

Risolviamo direttamente l'eliminazione di Gauss sulla matrice che ha come colonne le colonne di A e poi i termini noti. Se i termini non sono noti a priori possono esserci dei problemi, e si può applicare il metodo della potenza inversa, che serve per calcolare l'autovalore di modulo minimo.

Dopo aver fattorizzato $A = LU$, bisogna risolvere i sistemi $L\boldsymbol{y} = \boldsymbol{b}$ e poi $U\boldsymbol{x} = \boldsymbol{y}$. Spezziamo quindi il problema in due parti, separando la fattorizzazione della matrice dalla trattazione vera e propria della soluzione.

2.4.2 Descrizione dell'algoritmo

L'eliminazione gaussiana trasforma un sistema della forma $A\boldsymbol{x} = \boldsymbol{b}$ in un sistema della forma $U\boldsymbol{x} = \boldsymbol{c}$.

Supponiamo di essere al k-esimo passo di fattorizzazione, cioè supponiamo che la parte della matrice superiore alla riga A_k sia già stata ridotta a una triangolare superiore. A ogni passo, le operazioni sulle righe della matrice modificano anche il blocco quadrato in basso a destra, con le righe dalla $k+1$-esima all'n-esima.

Sottraiamo alla i-esima equazione la k-esima moltiplicata per un opportuno coefficiente.

Descrivendo l'algoritmo, per $k = 1, \ldots, n-1$ (passi di fattorizzazione), per $i = k+1, \ldots, n$ (righe successive alla k-esima), definisco l'elemento

$$M_{ik} = \frac{A_{ik}^{(k)}}{A_{kk}^{(k)}}$$

e sottraggo a $M_{ik} * A_k$, in modo che

$$A_{ik}^{(k+1)} = 0$$

Ad esempio, nel passo $k = 1$, prendiamo la prima riga e la sottraiamo a tutte le altre moltiplicata per il rispettivo coefficiente M_{ik}. Nel passo successivo lasciamo invariate le righe precedenti alla k-esima, e così via.

I coefficienti nel blocco quadrato degli A_{ij} con $k+1 < j < n, k+1 < i < n$ in basso a destra, che vengono modificati a ogni passo k, diventano:

$$A_{ij}^{(k+1)} = A_{ij}^{(k)} - M_{ik} * A_{kj}^{(k)}$$

Analogamente per il termine noto:

$$b_i^{(k+1)} = b_i^{(k)} - M_{ik} b_k^{(k)}$$

2.4.3 Matrice elementare di Gauss

◊ **Definizione** Sia M_k una matrice triangolare inferiore, con $\det M_k = 1$, divisa in quattro blocchi e tale che:

1. nel blocco $k-1 \times k-1$ in alto a sinistra c'è l'identità,
2. i blocchi di nord-est e di sud-ovest sono nulli,
3. la matrice in basso a destra ha la prima sottocolonna della forma

$$\begin{matrix} 1 \\ -M_{k+1,k} \\ -M_{k+2,k} \\ \cdots \\ -M_{n,k} \end{matrix}$$

dove gli M_{ik} sono i coefficienti definiti prima, mentre la prima riga di questo blocco è il vettore e 1 $\boldsymbol{e}_1$, e nel quadrato in basso c'è l'identità.

In forma sintetica:

I_{k-1}		0
	1	0 0
5 : 6		
0	$-M_{k+1,k}$ $-M_{n,k}$	I_k

In maniera compatta: $M_k = I - \boldsymbol{w}_k * \boldsymbol{e}_k^t$, dove $\boldsymbol{w}_k$ è un vettore colonna con le prime k componenti uguali a 0, e le restanti uguali a M_{ik} e $\boldsymbol{e}_k$ è il k-esimo vettore della base canonica. Il prodotto tra il vettore colonna e il vettore riga è una matrice.

2.4.4 Rappresentazione compatta dell'eliminazione di Gauss

Poniamo $A^{(1)} = A$, $A^{(k+1)} = M_k A^{(k)}$ per $k = 1, \ldots, n-1$, dove $A^{(k)}$ indica la matrice ottenuta al passo k-esimo. Ogni passo dell'eliminazione di Gauss può essere quindi rappresentato in modo compatto come un prodotto tra due matrici. Allora, chiamando U la matrice triangolare superiore ottenuta all'ultimo passo, si ha:

$$U = M_{n-1} \ldots M_2 M_1 A$$

Per poter scrivere $A = LU$, ci si chiede se il prodotto delle M_i è invertibile, e questo è vero perché è prodotto di matrici invertibili. Allora si può scrivere:

$$A = (M_{n-1} M_{n-2} \ldots M_2 M_1)^{-1} U$$

ed esplicitando la matrice inversa del prodotto:

$$A = M_1^{-1} M_2^{-1} \ldots M_{n-1}^{-1} U$$

Osservazione 2.3

Siccome L è la matrice triangolare inferiore con entrate uguali a 1 sulla diagonale, il suo determinante, prodotto delle entrate diagonali, è 1, e quindi

$$\det A = \det(LU) = \det L * \det U = \det U$$

allora la fattorizzazione della matrice può essere considerata anche come un modo per calcolare il determinante.

Teniamo conto del fatto che:

$$M_k = I - \boldsymbol{w}_k \boldsymbol{e}_k^t$$

Allora

$$M_k^{-1} = I + \boldsymbol{w}_k \boldsymbol{e}_k^t$$

infatti si verifica che

$$M_k^{-1}M_k = (I + \boldsymbol{w}_k\boldsymbol{e}_k^t)(I - \boldsymbol{w}_k\boldsymbol{e}_k^t) = I + \boldsymbol{w}_k\boldsymbol{e}_k^t - \boldsymbol{w}_k\boldsymbol{e}_k^t - \boldsymbol{w}_k\boldsymbol{e}_k^t\boldsymbol{w}_k\boldsymbol{e}_k^t$$

ma $\boldsymbol{e}_k^t\boldsymbol{w}_k$ è il vettore nullo, gli altri due termini sono opposti e rimane $M_k^{-1}M_k = I$. Consideriamo allora $L = M_1^{-1}M_2^{-1}\dots M_{n-1}^{-1}$, che si riscrive come

$$= (I + \boldsymbol{w}_1\boldsymbol{e}_1^t) * (I + \boldsymbol{w}_2\boldsymbol{e}_2^t)\dots(I + \boldsymbol{w}_{k-1}\boldsymbol{e}_{k-1}^t)$$

$$= I + \sum_{i=1}^{n-1} \boldsymbol{w}_i\boldsymbol{e}_i^t$$

L'ultimo passaggio vale perché, ad esempio

$$(I + \boldsymbol{w}_1\boldsymbol{e}_1^t)(I + \boldsymbol{w}_2\boldsymbol{e}_2^t) = I + \boldsymbol{w}_1\boldsymbol{e}_1^t + \boldsymbol{w}_2\boldsymbol{e}_2^t + \boldsymbol{w}_1\boldsymbol{e}_1^t\boldsymbol{w}_2\boldsymbol{e}_2^t$$

$\boldsymbol{e}_1^t\boldsymbol{w}_2 = 0$ come prima, perché hanno zeri in posizioni complementari. In generale

$$\boldsymbol{e}_i^t\boldsymbol{w}_k = 0 \forall i \leq k$$

i termini misti si annullano, e quindi rimane

$$(I + \boldsymbol{w}_1\boldsymbol{e}_1^t)(I + \boldsymbol{w}_2\boldsymbol{e}_2^t) = I + \boldsymbol{w}_1\boldsymbol{e}_1^t + \boldsymbol{w}_2\boldsymbol{e}_2^t$$

e, procedendo così fino al passo n-esimo, le colonne $\boldsymbol{w}_k\boldsymbol{e}_k^t$ si sommano una accanto all'altra, e si ottiene la matrice triangolare inferiore L cercata, con entrate uguali a 1 sulla diagonale, zeri nella parte superiore, e i moltiplicatori nella parte inferiore (i moltiplicatori in L sono positivi).

2.4.5 Operazioni sul termine noto

Le operazioni che l'eliminazione gaussiana fa sul termine noto equivalgono alla risoluzione del sistema $L\boldsymbol{y} = \boldsymbol{b}$. Eseguo, per $k = 1, \dots, n$ e per $i = k+1, \dots, n$:

$$y_i = b_i - L_{ik}y_k = b_i - M_{ik}y_k.$$

Esempio 2.6

Ad esempio, nel caso di una matrice 4×4,

$$y_1 = b_1$$

$$y_2 = b_2 - L_{21}y_1$$

$$y_3 = b_3 - L_{31}y_1 - L_{32}y_2$$

$$y_4 = b_4 - L_{41}y_1 - L_{42}y_2 - L_{43}y_3$$

Al primo passo di eliminazione gaussiana, il primo coefficiente è già definitivo mentre tutti gli altri vengono modificati. Al secondo passo anche il secondo coefficiente risulta definitivo, e così via. Al terzo passo, si lavora solo sull'ultima componente.

Si ha un aggiornamento progressivo per colonne.

La fattorizzazione ha un costo computazionale di n^3 operazioni, mentre la risoluzione vera e propria ne richiede n^2.

2.4.6 Riepilogo

Vogliamo risolvere il sistema $A\boldsymbol{x} = \boldsymbol{b}$, e nell'algoritmo dell'eliminazione gaussiana, per $k = 1, \ldots, n-1$, per $i = k+1, \ldots, n$, compiamo le seguenti operazioni:

1. definiamo i moltiplicatori

$$M_{ik} = \frac{|A_{ik}|}{|A_{kk}|}$$

2. per $j = k+1, \ldots, n$

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - M_{ik} a_{kj}^{(k)}$$

3. sul termine noto

$$b_i^{(k+1)} = b_i^{(k)} - M_{ik} b_k^{(k)}$$

4. passo quindi dal sistema $A\boldsymbol{x} = \boldsymbol{b}$ al sistema $U\boldsymbol{x} = \boldsymbol{c}$ con $\boldsymbol{c}$ soluzione di $L\boldsymbol{y} = \boldsymbol{b}$.

5. U è la matrice ottenuta dopo l'eliminazione gaussiana, mentre L ha entrate uguali a 1 sulla diagonale principale e ha i coefficienti M_{ik} nella parte inferiore.

2.4.7 Problema dell'esistenza della fattorizzazione

Data una matrice quadrata A, chiamiamo sottomatrici principali di A le matrici A_k, ottenute privando A della k-esima riga e della k-esima colonna. Si dicono matrici principali di testa le sottomatrici che contengono il coefficiente A_{11}.

Teorema 2.5

Data una matrice quadrata A in $\mathbb{C}_{n,n}$, se le matrici A_k sono non singolari per $k = 1 \ldots, n$, allora esiste unica la fattorizzazione $A = LU$.

Nel momento del calcolo del moltiplicatore M_{ik}, in aritmetica esatta si hanno problemi se $A_{kk} = 0$. Siccome la matrice è non singolare, esisterà una riga in cui il termine in posizione k è diverso da 0, allora, per evitare problemi, si scambia questa riga con la k-esima.

Teorema 2.6

Data $A \in \mathbb{C}_{n,n}$, allora esiste P matrice di permutazione tale che PA è fattorizzabile nel prodotto LU.

Una matrice di permutazione che scambia le righe i-esima e k-esima si ottiene scambiando tali righe nella matrice identica, in particolare, considerando i quattro coefficienti $A_{ii} = A_{kk} = 1$ e $A_{ik} = A_{ki} = 0$, si inverte il loro valore, ponendo $A_{ii} = A_{kk} = 0$ e $A_{ik} = A_{ki} = 1$.

2.5 Tecniche del pivot

2.5.1 Tecnica del pivot parziale

Supponiamo di essere al k-esimo passo di fattorizzazione, in cui bisogna azzerare i coefficienti della k-esima colonna a partire dall'entrata $k+1$. Cerchiamo l'indice $\bar{i}$ tale che $A_{\bar{i},k} = \max_{i=k}^{n} |a_{ik}|$, e scambiamo quindi la $\bar{i}$-esima riga con la k-esima,

applicando alla matrice $A^{(k)}$ la matrice di permutazione p_k, che è l'identità in cui scambio la $\bar{i}$-esima riga con la k-esima.

Bisogna verificare che anche moltiplicando per la matrice di permutazione, la fattorizzazione rimane del tipo LU.

2.5.2 Esempio in dimensione n=4

In dimensione $n = 4$, all'ultimo passo dell'eliminazione si ottiene

$$U = M_3 P_3 M_2 P_2 M_1 P_1 A,$$

(le matrici di permutazione sono alternate a quelle elementari)

Le matrici di permutazione hanno la proprietà che $P_k P_k = I$, allora, inserisco il doppio prodotto nell'uguaglianza nelle posizioni opportune, per far comparire vicine tra loro le matrici di permutazione.

Ad esempio:

$$\text{passo 1}, \quad U = M_3 P_3 M_2 P_2 M_1 P_2 P_2 P_1 A,$$

Ponendo $\bar{M}_1 = P_2 M_1 P_2$, al passo successivo otteniamo:

$$U = M_3 P_3 M_2 P_3 P_3 \bar{M}_1 P_3 P_3 P_2 P_1 A$$

e ponendo $\bar{M}_2 = P_3 M_2 P_3$, e $\hat{M}_1 = P_3 \bar{M}_1 P_3$, ottengo:

$$U = M_3 \bar{M}_2 \hat{M}_1 P_3 P_2 P_1 A$$

e si pone $P_3 P_2 P_1 A = PA$.

Vogliamo verificare che anche le matrici segnate sono ancora matrici elementari di Gauss, e quindi invertibili.

Consideriamo, ad esempio, $\bar{M}_2 = P_3 M_2 P_3$

$$M_2 = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & M_{32} & 1 & 0 \\ 0 & M_{42} & 0 & 1 \end{matrix}$$

Invece

$$P_3 = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{matrix}$$

Allora calcoliamo il prodotto $P_3 M_2 P_3$:

$$P_3 M_2 = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & M_{42} & 0 & 1 \\ 0 & M_{32} & 1 & 0 \end{matrix}$$

(scambiamo la terza riga con la quarta). Invece, moltiplicando a destra per P_3, scambiamo la terza e la quarta colonna, e otteniamo la matrice

$$\bar{M}_2 = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & M_{42} & 1 & 0 \\ 0 & M_{32} & 0 & 1 \end{matrix}$$

che ha la stessa forma di M_2.

Consideriamo invece la matrice P_2 che scambia la seconda riga con la terza:

$$p_2 = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Allora

$$P_2 M_2 = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & M_{32} & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & M_{42} & 0 & 1 \end{matrix}$$

e moltiplicando per P_2:

$$P_2 M_2 P_2 = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & M_{32} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & M_{42} & 1 \end{matrix}$$

che non è più una matrice elementare di Gauss.

In ogni caso, la moltiplicazione per P_2 non è più un'operazione ammissibile nell'eliminazione di Gauss al passo $k = 2$, perché si può operare solo nella sottomatrice 2×2 in basso a destra, quindi non ci sono problemi.

2.5.3 Fattorizzazione

Dato il sistema $A\boldsymbol{x} = \boldsymbol{b}$, moltiplichiamo per P ambo i membri per far comparire la matrice di permutazione, e otteniamo:

$$PA\boldsymbol{x} = P\boldsymbol{b}$$

$$PA = LU \quad \longrightarrow \quad LU\boldsymbol{x} = P\boldsymbol{b}$$

Allora bisogna considerare il termine noto permutato $\bar{\boldsymbol{b}}$, e risolviamo prima il sistema lineare

$$L\boldsymbol{y} = \bar{\boldsymbol{b}}$$

e poi

$$U\boldsymbol{x} = \boldsymbol{y}$$

2.5.4 Effetti del pivot parziale

Il pivot parziale ha un effetto stabilizzante, ovvero riduce l'errore algoritmico. In particolare:

1. Come prima, L è una matrice triangolare inferiore con 1 sulla diagonale principale e M_{ik} nella parte inferiore.

2. I moltiplicatori hanno espressione

$$M_{ik} = \frac{|A_{ik}|}{\max_{j=k}^{n} |A_{jk}|}$$

quindi, per ogni $i = k+1, \dots, n$ i coefficienti sono ≤ 1.

3. La crescita degli elementi di U è limitata, infatti i coefficienti della "nuova" matrice sono:

$$A_{ij}^{(k+1)} = A_{ij}^{(k)} - M_{ik} A_{kj}^{(k)}$$

mentre per $i, j = 1, \dots k$ gli elementi della matrice rimangono invariati, e quindi

$$A_{ij}^{(k+1)} = A_{ij}^{(k)}$$

in definitiva

$$|A_{ij}|^{(k+1)} \leq |A_{ij}|^{(k)} + |M_{ik}||A_{kj}|^{(k)} \leq |A_{ij}| + |A_{kj}|$$

Chiamiamo $A_m^{(k)}$ il massimo modulo degli elementi della matrice $A^{(k)}$. Allora, passando ai massimi nella disuguaglianza sopra,

$$|A_m^{(k+1)}| \leq |A_m^{(k)}| + |A_m^{(k)}| = 2|A_m^{(k)}|$$

e applicando a ritroso la relazione

$$|A_m|^{(n)} \leq 2^{n-1} |A_m^{(1)}|$$

2.5.5 Errore algoritmico

Teorema 2.7

Sia $A \in \mathbb{C}_{n,n}$ tale che i suoi elementi siano numeri macchina (focus sull'errore algoritmico). Chiamiamo $\bar{L}, \bar{U}$ le matrici effettivamente calcolate, allora $\bar{L} * \bar{U} = A + \delta_A$ dove vale la relazione

$$|\delta_A| \leq 2nu(|A| + |\bar{L}| * |\bar{U}|) + O(u^2)$$

(a livello notazionale, chiamiamo $|B|$ la matrice che ha come componenti i moduli degli elementi di B)

In questa relazione, oltre alla dipendenza dalla dimensione, da A e dalla precisione di macchina, il fatto significativo è che compaiono $\bar{L}, \bar{U}$.

Teorema 2.8

Siano $A, B \in \mathbb{C}_{n,n}$ e, considerando $\bar{L}, \bar{U}$ effettivamente calcolate, chiamiamo $\bar{\boldsymbol{y}}$ la soluzione del sistema $\bar{L}\boldsymbol{y} = \boldsymbol{b}$ e $\bar{\boldsymbol{x}}$ soluzione di $\bar{U}\boldsymbol{x} = \boldsymbol{y}$. Allora $\bar{\boldsymbol{x}}$ è la soluzione di

$$(A + \delta_A)\bar{\boldsymbol{x}} = \boldsymbol{b}$$

con

$$\delta_A \leq 4nu(|A| + |\bar{L}||\bar{U}| + o(u^2))$$

(interpretiamo l'errore algoritmico come errore inerente del problema perturbato)

Esempio 2.7

Consideriamo la matrice

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

$$\boldsymbol{b} = (1 + \varepsilon, 1)$$

dove ε è scelto in modo che la soluzione del sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$ sia $(1, 1)$. Il problema è ben condizionato, infatti

$$A^{-1} = \begin{matrix} 0 & 1 \\ 1 & -\varepsilon \end{matrix}$$

$$K(A) = |A|_\infty * |A^{-1}|_\infty = (1 + \varepsilon)^2$$

e quindi se ε è piccolo, $K(A) \simeq 1$.

Per quanto riguarda la stabilità, senza tecnica del pivot:

$$L = \begin{matrix} 1 & 0 \\ 1/\varepsilon & 1 \end{matrix}$$

$$U = \begin{matrix} \varepsilon & 1 \\ 0 & -1/\varepsilon \end{matrix}$$

$$\bar{\boldsymbol{x}} = (0, 1)$$

mentre la soluzione esatta ha entrambe le componenti uguali a 1, e il problema è dato dalla crescita dei coefficienti di U.

2.5.6 Tecnica del pivot totale

Per ogni passo k, cerchiamo $\bar{i}, \bar{j}$ tale che

$$|A_{\bar{i},\bar{j}}| = \max_{i,j=k}^{n} |A_{i,j}|$$

assumiamo di poter fare permutazioni di colonne $\hat{P}_k$ oltre che permutazioni di righe P_k.

Il problema che sorge è che, facendo permutazioni di colonne, si permuta l'ordine delle incognite.

Esempio 2.8

Nella catena di moltiplicazioni di matrici elementari, nel caso $n = 4$,

$$U = M_3 P_3 M_2 P_2 M_1 P_1 A \hat{P}_1 \hat{P}_2 \hat{P}_3$$

ma poniamo $\hat{P}_1 \hat{P}_2 \hat{P}_3 = \hat{P}$. L'altro pezzo di U viene trattato come prima.

Come prima

$$PA\hat{P} = LU$$

e per risolvere il sistema $A\boldsymbol{x} = \boldsymbol{b}$ permutiamo a sinistra,

$$PA\boldsymbol{x} = P\boldsymbol{b}$$

bisogna inserire la $\hat{P}$, allora poniamo $\hat{P}\boldsymbol{z} = \boldsymbol{x}$

$$PA\hat{P}\boldsymbol{z} = P\boldsymbol{b}$$

Calcoliamo $\bar{\boldsymbol{b}}$ permutazione del termine noto, e risolviamo quindi:

$$L\boldsymbol{y} = \bar{\boldsymbol{b}}$$

$$U\boldsymbol{z} = \boldsymbol{y}$$

$$\hat{P}\boldsymbol{z} = \boldsymbol{x}, \quad \text{è una moltiplicazione matrice per vettore}$$

Questo procedimento ha un effetto più stabilizzante rispetto al pivot parziale, infatti si ha:

$$A_k \leq \sqrt{k \prod_{j=2}^{n} j^{j-1}}$$

e non sono state trovate matrici per cui vale l'uguaglianza.

2.5.7 Costo computazionale

La risoluzione *forward* o *backward*, considerando il numero di operazioni moltiplicative, ha come costo computazionale

$$\sum_{i=1}^{n} i$$

infatti, per ogni incognita, si ha l'espressione

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j}{a_{ii}}$$

Invece la fattorizzazione LU ha costo

$$\sum_{k=1}^{n-1} [(\sum_{i=k+1}^{n} 1) + \sum_{i=k+1}^{n} \sum_{j=k+1}^{n} 1]$$

che è dell'ordine di $n^3/3$ (il primo addendo serve per il calcolo dei moltiplicatori, e il secondo per il calcolo del blocco quadrato in basso a destra).

2.6 Fattorizzazione di Cholesky

2.6.1 Esistenza della fattorizzazione $A = R^t R$

Questo tipo di fattorizzazione si applica solo al caso di matrici definite positive.

Teorema 2.9

Supponiamo di avere $A \in \mathbb{R}^{n\times n}$ simmetrica e non singolare. A è definita positiva se e solo se esiste $R \in \mathbb{R}^{n\times n}$ triangolare superiore tale che $A = R^t R$.

Dimostrazione

$2 \longrightarrow 1$: A è non singolare, e supponiamo $A = R^t R$, allora R è non singolare infatti

$$\det A = \det R^t * \det R = |\det R|^2$$

e quindi $\det A \neq 0$ implica R non singolare. Allora

$$\forall \boldsymbol{x} \neq 0, \quad \boldsymbol{y} = R\boldsymbol{x} \neq 0,$$

quindi

$$|\boldsymbol{y}|_2^2 \geq 0$$

e siccome ho escluso il caso nullo,

$$|\boldsymbol{y}|_2^2 > 0.$$

e A è definita positiva.

$1 \longrightarrow 2$: consideriamo A simmetrica e definita positiva:

$$\begin{pmatrix} \alpha & \boldsymbol{a}^t \\ \boldsymbol{a} & A_* \end{pmatrix}$$

dove A_* è una sottomatrice di dimensione $n-1 \times n-1$ e a $\boldsymbol{a}$ è un vettore di $n-1$ componenti.

Definiamo R, R^t nel seguente modo:

$$R = \begin{pmatrix} \rho & \boldsymbol{r}^t \\ \boldsymbol{0} & R_* \end{pmatrix}$$

$$R^t = \begin{pmatrix} \rho & \boldsymbol{0}^t \\ \boldsymbol{r} & (R_*)^t \end{pmatrix}$$

Il prodotto delle due matrici è:

$$R^t R = \begin{pmatrix} \rho^2 & \rho * \boldsymbol{r}^t \\ \rho * \boldsymbol{r} & \boldsymbol{r}\boldsymbol{r}^t + R_*^t R_* \end{pmatrix}$$

ed eguagliamo termine a termine la decomposizione a blocchi con il risultato del prodotto ottengo:

$$\alpha = \rho^2 \longrightarrow \rho = \sqrt{\alpha}$$
$$\boldsymbol{a} = \rho \boldsymbol{r}\boldsymbol{r} = \frac{1}{\sqrt{\alpha}}\boldsymbol{a}$$
$$\boldsymbol{a}^t = \rho \boldsymbol{r}^t \longrightarrow \boldsymbol{r}^t = \frac{\boldsymbol{a}^t}{\sqrt{\alpha}}$$
$$A_* = \boldsymbol{r}\boldsymbol{r}^t + R_*^t R_* \longrightarrow R_*^t R_* = A_* - \boldsymbol{r}\boldsymbol{r}^t = A_* - 1/\alpha \boldsymbol{a}\boldsymbol{a}^t$$

Definiamo la quantità

$$\hat{A}_* = R_*^t R_*$$

Se $\hat{A}_*$ è ancora simmetrica e definita positiva, si può iterare la procedura, lavorando su $R_*^t R_*$, fino a ottenere la fattorizzazione con R triangolare superiore. Quindi verifichiamo che $\hat{A}_*$ è simmetrica definita positiva.

Simmetria: sappiamo che

$$\hat{A}_*^t = (R_*^t R_*)^t = R_*^t (R_*^t)^t = R_*^t R_* = \hat{A}_*$$

Positività: appliciamo ad A una matrice elementare M per azzerare la prima sottocolonna, con

$$A = \begin{pmatrix} \alpha & \boldsymbol{a}^t \\ \boldsymbol{a} & A_* \end{pmatrix}$$

$$M = \begin{pmatrix} 1 & 0 \\ -\boldsymbol{a}/\alpha & I \end{pmatrix}$$

Il risultato del prodotto è:

$$MA = \begin{pmatrix} \alpha & \boldsymbol{a}^t \\ 0 & -1/\alpha \boldsymbol{a}\boldsymbol{a}^t + A_* \end{pmatrix}$$

(come nell'eliminazione di Gauss ordinaria, la prima riga rimane invariata e la prima colonna viene azzerata)

Per poter azzerare una sottoriga, applichiamo alla matrice MA un passo di eliminazione gaussiana per righe. Eseguiamo il prodotto $MA\hat{M}$ con

$$MA = \begin{pmatrix} \alpha & \boldsymbol{a}^t \\ 0 & A_* - 1/\alpha \boldsymbol{a}\boldsymbol{a}^t \end{pmatrix}$$

$$\hat{M} = \begin{pmatrix} 1 & -\boldsymbol{a}^t/\alpha \\ 0 & I \end{pmatrix}$$

(ora la sottocolonna dei coefficienti è sulla prima riga)

Calcolando il prodotto

$$MA\hat{M} = \begin{pmatrix} \alpha & 0 \\ 0 & A_* - 1/\alpha \boldsymbol{a}\boldsymbol{a}^t \end{pmatrix}$$

Osserviamo che $\hat{M} = M^t$. Ponendo $\boldsymbol{y} = M^t \boldsymbol{x}$, per ogni $\boldsymbol{x} \neq 0$,

$$\boldsymbol{x}^t MA\hat{M}\boldsymbol{x} = \boldsymbol{x}^t MAM^t \boldsymbol{x} = \boldsymbol{y}^t A \boldsymbol{y}$$

e $\boldsymbol{y} = M^t\boldsymbol{x} \neq 0 \forall \boldsymbol{x} \neq 0$ perché M è non singolare. Siccome per ipotesi A è definita positiva, $\boldsymbol{y}^t A\boldsymbol{y}$ è maggiore di 0, e quindi la matrice grande $MA\hat{M}$ è definita positiva.

Allora

$$A_* - 1/\alpha \boldsymbol{a}\boldsymbol{a}^t$$

è definita positiva, perché sottomatrice principale di una definita positiva. Quindi $\hat{A}_* = R_*^t R$ è definita positiva e la procedura si può iterare sulla matrice di dimensioni $n-1 \times n-1$. Lavorando su $R_*^t R_*$, si determinano progressivamente gli elementi di R_*, triangolare superiore.

cvd

Osservazione 2.4

Consideriamo

$$A_* - 1/\alpha \boldsymbol{a}\boldsymbol{a}^t.$$

Alla matrice A_* definita positiva, sottraiamo una matrice semidefinita positiva, e non è ovvio che il risultato sia una definita positiva. Infatti, ad esempio

$$I - (0,1)(0,1)^t$$

non è definita positiva.

2.6.2 Complemento di Shur

Questa dimostrazione permette di osservare la seguente proprietà: il complemento di Shur di una definita positiva è definito positivo, dove $\hat{A}_*$ è il complemento di Shur di α in A.

◊ **Definizione** In generale, data una matrice $\bar{A}$ a blocchi della forma

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

dove il primo blocco è $n \times n$, il secondo $n \times m$, e così via, e dove la dimensione della matrice è $(n+m) \times (n+m)$, si chiede che A_{11} sia non singolare, allora il **complemento di Shur** è definito come

$$A_{22} - A_{21}A_{11}^{-1}A_{12}$$

2.6.3 Algoritmo per il calcolo di R

In questa scrittura supponiamo che gli elementi di A vengano progressivamente sovrascritti con quelli di R trovati passo per passo.

fork=1:n

$$A(k,k) = \sqrt{A(k,k)}$$

end

$$fork = 1 : n-1$$

$$forj = k+1 : n$$

$$A(k,j) = A(k,j)/A(k,k)$$

$$A(j,k) = 0$$

end

fori=k+1:n

forj=k+1:n

$$A(i,j) = A(i,j) - A(i,k) * A(j,k)$$

end

end

end

Descrizione esplicita: per $k = 1 : n$ calcolo

$$R_{kk} = A_{kk}^{(k+1)} = \sqrt{A_{kk}} = \rho$$

e al passo n-esimo questa è l'unica operazione da fare.

Allora, per $k = 1 : n-1$, facciamo le seguenti operazioni:

1. per $j = k+1 : n$,

$$R_{kj} = A_{kj}^{(k+1)} = \frac{A_{kj}}{A_{kk}} = \frac{A_{kj}}{R_{kk}}$$

(queste istruzioni corrispondono a $\frac{\boldsymbol{a}}{\sqrt{\alpha}}$, definizione del vettore $\boldsymbol{r}$)

2. per $i = k+1 : n$, e per $j = k+1 : n$,

$$\hat{A}_{ij,*} = A_{ij}^{(k+1)} = A_{ij} - A_{ki}A_{kj} = A_{ij} - R_{ki}R_{kj}$$

(nell'ultimo blocco di istruzioni si definiscono i coefficienti di $\hat{A}_*$, cioè $A_* - \boldsymbol{r}\boldsymbol{r}^t$).

Ovviamente, alla fine di ogni passo k, le entrate A_{kj} con $j < k$ vengono annullate.

2.6.4 Costo computazionale

È sufficiente calcolare solo metà della sottomatrice $A(k+1:n, k+1:n)$, che è simmetrica.

Le operazioni richieste sono quindi:

1. n estrazioni di radici quadrate (una per ogni passo di fattorizzazione);

2. $n-k$ operazioni moltiplicative per ogni passo k. Infatti, al passo $\boldsymbol{r}^t$ ha indici da $k+1$ a n, eper ciascun elemento houna divisione;

3. per ogni elemento di $\hat{A}_*$, ho un'operazione moltiplicativa, e quindi per ogni passo le operazioni sono

$$\sum_{i=1}^{m} i = m(m+1)/2$$

con $m = n-k+1-1 = n-k$, cioè $m(m+1)/2 = (n-k)(n-k+1)/2$

Complessivamente il numero di operazioni è:

$$n + \sum_{k=1}^{n-1} [n-k+(n-k)(n-k+1)/2]$$

e raccogliendo $n-k$ otteniamo:

$$n + \sum_{k=1}^{n-1} (n-k)(1+(n-k+1)/2)$$

$$n + 1/2 \sum_{k=1}^{n-1} (n-k)(2+n-k+1)$$

$$n + 1/2 \sum_{k=1}^{n-1} (n-k)(n-k+3)$$

$$\simeq n + 1/2 \sum_{k=1}^{n-1} (n-k)^2$$

$$= n^3/6$$

Il costo computazionale è pari a quello dell'eliminazione di Gauss per le simmetriche.

2.6.5 Stabilità

Considerando il prodotto $A = R^t R$, segue che

$$A_{kk} = \sum_{j=1}^{n} R^t_{kj} R_{jk}$$

$$A_{kk} = \sum_{j=1}^{n} R^2_{jk}$$

Allora ciascun R_{jk} è tale che $R^2_{jk} \leq A_{kk}$, quindi

$$|R_{jk}| \leq \sqrt{A_{kk}} \, \forall k$$

Chiamando R_m l'elemento di modulo massimo, si ha

$$R_m \leq \sqrt{A_m}$$

Quindi gli elementi di R hanno crescita controllata da $\sqrt{A_m}$.
L'algoritmo è stabile.

2.6.6 Rapporto tra fattorizzazione LU e di Cholesky

Data una matrice simmetrica definita positiva, consideriamo il rapporto tra la fattorizzazione LU e quella di Cholesky. La fattorizzazione LU esiste unica per una definita positiva, perché le matrici di testa di una definita positiva sono definite positive, e non singolari, quindi le ipotesi del teorema di esistenza e unicità della fattorizzazione sono soddisfatte.

Tuttavia $L \neq U^t$ perché la diagonale di L ha entrate uguali a 1.

Poniamo $U = D * \bar{U}$ dove D è una matrice con elementi diagonali dati dalla diagonale principale di U.

Per ipotesi

$$A = LU = A^t = (LD\bar{U})^t = \bar{U}^t D L^t$$

e per l'unicità di fattorizzazione,

$$LU = \bar{U}^t D L^t \quad \longrightarrow \quad L = \bar{U}^t, \quad DL^t = U$$

allora

$$A = \bar{U}^t D \bar{U}$$

con D matrice a elementi diagonali positivi. Poniamo $\boldsymbol{x} = \bar{U}\boldsymbol{y}$, allora

$$\boldsymbol{x}^t D \boldsymbol{x} = (\bar{U}\boldsymbol{y})^t D (\bar{U}\boldsymbol{y})$$

$$= \boldsymbol{y}^t \bar{U}^t D \bar{U} \boldsymbol{y} = \boldsymbol{x}^t A \boldsymbol{x} > 0$$

perché A è definita positiva, allora $\bar{U}$ è non singolare.

$$A = \bar{U}^t D \bar{U}$$

Definiamo $D^{1/2}$ come la matrice che ha come elementi diagonali le radici quadrate degli elementi di D.

Allora

$$A = \bar{U}^t D^{1/2} D^{1/2} \bar{U} = (D^{1/2}\bar{U})^t D^{1/2} \bar{U} = R^t R$$

cioè

$$R = D^{1/2}\bar{U}$$

dove $\bar{U} = D^{-1}U$, $R = D^{1/2}D^{-1}U = D^{-1/2}U$.

Proposizione 2.1

L'eliminazione gaussiana conserva simmetria e definita positività.

Dim.

Simmetria: sappiamo che

$$A_{ij}^{(k+1)} = A_{ij}^{(k)} - M_{ik}A_{kj}^{(k)}$$

con

$$M_{ik} = \frac{A_{ik}^{(k)}}{A_{kk}^{(k)}}$$

cioè, sostituendo esplicitamente M_{ik}:

$$A_{ij}^{(k+1)} = A_{ij}^{(k)} - \frac{A_{ik}^{(k)}}{A_{kk}^{(k)}}A_{kj}^{(k)}$$

$A_{ji}^{(k)} = A_{ij}^{(k)}$ per simmetria di A, quindi si può scrivere:

$$A_{ij}^{(k+1)} = A_{ji}^{(k)} - \frac{A_{ki}^{(k)}}{A_{kk}^{(k)}}A_{jk}^{(k)}$$

e tenendo conto che $\frac{A_{jk}}{A_{kk}} = M_{jk}$ si ha

$$A_{ij}^{(k+1)} = A_{ji}^{(k)} - M_{jk}A_{ki}^{(k)} = A_{ji}^{(k+1)}$$

cioè la sottomatrice in basso a destra ottenuta dopo il passo k-esimo è ancora simmetrica.

Definita positività: consideriamo

$$A = \begin{pmatrix} \alpha & \boldsymbol{a}^t \\ \boldsymbol{a} & A_* \end{pmatrix}$$

$$M = \begin{pmatrix} 1 & 0 \\ -\boldsymbol{a}/\alpha & I \end{pmatrix}$$

$$MA = \begin{pmatrix} \alpha & \boldsymbol{a}^t \\ 0 & A_* - 1/\alpha\boldsymbol{a}\boldsymbol{a}^t \end{pmatrix}$$

L'ultima entrata è il complemento di Shur di α in A. Siccome A è definita positiva, anche il completamento è definito positivo. Allora l'eliminazione gaussiana è stabile sulle definite positive.

cvd

2.7 Fattorizzazione QR

2.7.1 Motivazioni

Q è una matrice unitaria, cioè $QQ^h = Q^hQ = 1$, allora $Q^{-1} = Q^h$, e R è una triangolare superiore.

Ci chiediamo per quale motivo serve un'alternativa alla fattorizzazione $PA = LU$.

Il vantaggio della fattorizzazione QR è una maggior stabilità rispetto alla $PA = LU$. Infatti la QR è una fattorizzazione con matrici elementari unitarie, che conservano la norma 2.

Lo svantaggio è il raddoppio del costo computazionale.

Non occorre la tecnica del pivot.

Ci sono situazioni più sofisticate in cui si usa una tecnica di pivot per colonne, per garantire che gli elementi diagonali della R abbiano una certa proprietà di ordinamento.

Questa fattorizzazione si usa solo in casi limite, ad esempio nel calcolo degli autovalori (spettro completo) e del rango, e della s.v.d.

Data una matrice A hermitiana, allora si può diagonalizzare con trasformazioni unitarie, scrivendo $Q\lambda Q^h$.

La decomposizione in fattori singolari è l'estensione di quest'idea anche ai casi in cui la matrice non è simmetrica e non è quadrata.

La fattorizzazione QR esiste sempre ma non è unica.

2.7.2 Risoluzione del sistema

Con la fattorizzazione QR A viene fattorizzata nel prodotto di una matrice unitaria e di una triangolare superiore.

Q è unitaria quindi $Q^{-1} = Q^h$.

Nel sistema $A\boldsymbol{x} = \boldsymbol{b}$, cioè $QR\boldsymbol{x} = \boldsymbol{b}$, poniamo $R\boldsymbol{x} = \boldsymbol{y}$, risolvo prima

$$Q\boldsymbol{y} = \boldsymbol{b} \longrightarrow \boldsymbol{y} = Q^{-1}\boldsymbol{b} = Q^h\boldsymbol{b} \quad \text{prodotto matrice per vettore}$$

quindi si risolve

$$R\boldsymbol{x} = \boldsymbol{y}$$

con R matrice triangolare.

Il costo di risoluzione è $2/3n^3$.

2.7.3 Matrici elementari di Hausolder

definizione|Una matrice elementare è una matrice della forma $E = I - \sigma\boldsymbol{u}\boldsymbol{v}^h$, con $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{C}^n$, σ parametro, e il prodotto $\boldsymbol{u}\boldsymbol{v}^h$ è una diade.

La matrice della diade ha rango uguale a 1.

Nella fattorizzazione QR consideriamo le matrici elementari di Hausolder (o matrici di riflessione). Queste matrici sono della forma:

$$P = I - \beta\boldsymbol{v}\boldsymbol{v}^h, \quad \beta \in \mathbb{R},\ \boldsymbol{v} \in \mathbb{C}^n$$

Proposizione 2.2

Le matrici di riflessione sono hermitiane, e sono unitarie se scegliamo $\beta = \frac{2}{|\boldsymbol{v}|_2}$.

Dim.

Hermitiane:

$$P^h = I - \beta(\boldsymbol{v}\boldsymbol{v}^h)^h = I - \beta\boldsymbol{v}\boldsymbol{v}^h = P$$

Unitarie: sappiamo che $I = PP^h$, e per hermitianità

$$PP^h = PP = (I - \beta \boldsymbol{v}\boldsymbol{v}^h)(I - \beta \boldsymbol{v}\boldsymbol{v}^h)$$

$$= I - \beta \boldsymbol{v}\boldsymbol{v}^h - \beta \boldsymbol{v}\boldsymbol{v}^h + \beta^2 \boldsymbol{v}\boldsymbol{v}^h \boldsymbol{v}\boldsymbol{v}^h$$

$$= I - 2\beta \boldsymbol{v}\boldsymbol{v}^h + \beta^2 \boldsymbol{v}(|\boldsymbol{v}|)^2 \boldsymbol{v}^h$$

(infatti $\boldsymbol{v}^h \boldsymbol{v} = (|\boldsymbol{v}|_2)^2$)
quindi

$$I = I - 2\beta \boldsymbol{v}\boldsymbol{v}^h + \beta(|\boldsymbol{v}|_2)^2 \boldsymbol{v}\boldsymbol{v}^h$$

implica

$$\beta \boldsymbol{v}\boldsymbol{v}^h * [-2 + \beta|\boldsymbol{v}|_2^2] = 0$$

cioè, se $\beta \neq 0$, deve valere $-2 + \beta|\boldsymbol{v}|_2^2 = 0$, cioè

$$\beta = \frac{2}{(|\boldsymbol{v}|_2)^2}$$

cvd

2.7.4 Altre proprietà sulle matrici di Hausolder

Sia $P = P(\boldsymbol{v})$ (la matrice di Hausolder è univocamente determinata da $\boldsymbol{v}$), con $P = I - \beta \boldsymbol{v}\boldsymbol{v}^h$, e $\beta = \frac{2}{(|\boldsymbol{v}|_2)^2}$. Allora per ogni $\boldsymbol{x} \in \boldsymbol{v}_\perp$, la matrice di Hausolder si comporta come l'identità. Infatti:

$$\boldsymbol{x}(I - \beta \boldsymbol{v}\boldsymbol{v}^h) = \boldsymbol{x} - \beta \boldsymbol{v}\boldsymbol{v}^h \boldsymbol{x} = \boldsymbol{x}$$

infatti $\boldsymbol{x}\boldsymbol{v} = 0$ per $\boldsymbol{x} \in \boldsymbol{v}_\perp$.

P applicata nel sottospazio span$\{\boldsymbol{v}\}$ è:

$$(I - \beta \boldsymbol{v}\boldsymbol{v}^h)(\gamma \boldsymbol{v}) = \gamma \boldsymbol{v} - \beta\gamma \boldsymbol{v}\boldsymbol{v}^h \boldsymbol{v}$$

$$= \gamma \boldsymbol{v} - \beta\gamma|\boldsymbol{v}|^2 \boldsymbol{v}$$

e per definizione di β:

$$= \gamma \boldsymbol{v} - \frac{2}{(|\boldsymbol{v}|_2)^2}\gamma|\boldsymbol{v}|^2 \boldsymbol{v}$$

$$= \gamma \boldsymbol{v} - 2\gamma \boldsymbol{v} = -\gamma \boldsymbol{v}$$

cioè si ha una riflessione.

2.7.5 Esistenza della matrice elementare P

Teorema 2.10

Per ogni $\boldsymbol{x} \in \mathbb{C}^n \setminus 0$ esiste $P \in \mathbb{C}_{n\times n}$ matrice elementare di Hausolder tale che $P\boldsymbol{x} = \alpha \boldsymbol{e}_1$.

Dimostrazione

Dobbiamo determinare il parametro $\alpha \in \mathbb{C}$ e la matrice elementare di Hausolder P.

Siccome $\alpha \in \mathbb{C}$, $\alpha = |\alpha| * e^{i\theta}$ (relazione $\circ$). Deve essere $P\boldsymbol{x} = \alpha \boldsymbol{e}_1$, allora

$$|P\boldsymbol{x}| = |\alpha||\boldsymbol{e}_1|_2 = |\alpha|$$

Inoltre P è un'isometria quindi conserva la norma 2, cioè:

$$(|P\boldsymbol{x}|_2)^2 = P^2\boldsymbol{x}^h\boldsymbol{x} = \boldsymbol{x}^h\boldsymbol{x} = (|\boldsymbol{x}|_2)^2$$

allora, eguagliando le due espressioni trovate per $(|P\boldsymbol{x}|_2)^2$, ottengo:

$$|\alpha| = |\boldsymbol{x}|_2$$

Per determinare l'angolo θ, dev'essere

$$\boldsymbol{x}^h P\boldsymbol{x} = \boldsymbol{x}^h \alpha \boldsymbol{e}_1 = \alpha \boldsymbol{x}^h \boldsymbol{e}_1 = \alpha \bar{x}_1 \quad \text{è uno scalare}$$

Per hermitianità di P:

$$\bar{(}\boldsymbol{x}^h P\boldsymbol{x}) = (\boldsymbol{x}^h P\boldsymbol{x})^h = \boldsymbol{x}^h P^h \boldsymbol{x} = \boldsymbol{x}^h P\boldsymbol{x}$$

allora siccome lo scalare è uguale al suo coniugato, è reale, quindi $\alpha\bar{x}_1 \in \mathbb{R}$.

$\bar{x}_1 \in \mathbb{C}$, allora possiamo scriverlo come $|x_1|e^{i\phi}$ (relazione $\circ\circ$).

$$\alpha\bar{x}_1 = |\alpha|e^{i\theta}|x_1|e^{i\phi}$$

e sostituendo l'espressione di α:

$$= |\boldsymbol{x}|_2|x_1|e^{i(\theta-\phi)}$$

$\alpha\bar{x}_1 \in \mathbb{R}$, allora

$$\theta - \phi = k\pi$$

ovvero $\theta = \phi + \delta\pi$ con $\delta = 0, 1$.

In conclusione:

$$\alpha = |\alpha|e^{i\theta} = |\boldsymbol{x}|_2 * e^{i(\phi+\delta\pi)}$$

$$e^{i\delta\pi} = \pm 1 \longrightarrow \alpha = \pm|\boldsymbol{x}|_2 e^{i\phi}$$

e per la $\circ\circ$,

$$= \pm|\boldsymbol{x}|_2 * \frac{x_1}{|x_1|}$$

Se $x_1 = 0$, allora poniamo $\alpha = -|\boldsymbol{x}|_2$.

Vale che

$$P\boldsymbol{x} = (I - \beta\boldsymbol{v}\boldsymbol{v}^h)(\boldsymbol{x}) = \boldsymbol{x} - \beta\boldsymbol{v}\boldsymbol{v}^h\boldsymbol{x},$$

e per definizione di β otteniamo:

$$P\boldsymbol{x} = \boldsymbol{x} - \frac{2}{(|\boldsymbol{v}|_2)^2}\boldsymbol{v}^h\boldsymbol{x}\boldsymbol{v}$$

Chiamiamo $\hat{\beta}$ il coefficiente che moltiplica $\boldsymbol{v}$, e mettiamo in evidenza $\hat{\beta}\boldsymbol{v}$:

$$\hat{\beta}\boldsymbol{v} = \boldsymbol{x} - P\boldsymbol{x} = \boldsymbol{x} - \alpha\boldsymbol{e}_1$$

Osservazione 2.5

A patto di escludere che $\hat{\beta} = 0$, è lecito porre $\hat{\beta} = 1$, in quanto

$$\begin{aligned} P(\gamma\boldsymbol{v}) &= I - \frac{2}{|\gamma\boldsymbol{v}|^2} * (\gamma\boldsymbol{v}) * (\gamma\boldsymbol{v}^h) \\ &= I - \frac{2}{|\gamma|^2|\boldsymbol{v}|_2}|\gamma|^2 * \boldsymbol{v}\boldsymbol{v}^h \\ &= I - \frac{2}{|\boldsymbol{v}|_2} * \boldsymbol{v}\boldsymbol{v}_h = P(\boldsymbol{v}) \end{aligned}$$

allora per il fatto che $P(\gamma\boldsymbol{v}) = P(\boldsymbol{v})$, possiamo porre $\hat{\beta} = 1$.

Tornando alla formula:

$$\hat{\beta}\boldsymbol{v} = \boldsymbol{x} - \alpha\boldsymbol{e}_1$$

e ponendo $\hat{\beta} = 1$, otteniamo:

$$\boldsymbol{v} = \boldsymbol{x} - \alpha\boldsymbol{e}_1$$

Nota bene: possiamo escludere il caso $\hat{\beta} = 0$, infatti, per definizione, $\hat{\beta} = 0$ se e solo se $\boldsymbol{x} \in \boldsymbol{v}_\perp$, e possiamo scegliere il vettore $\boldsymbol{x}$ al di fuori del complemento ortogonale.

Calcoliamo:

$$\boldsymbol{v}^h\boldsymbol{v} = \boldsymbol{v}^h(\boldsymbol{x} - \alpha\boldsymbol{e}_1) = (|\boldsymbol{x}|)^2 + |x_1||\boldsymbol{x}|_2$$

e la quantità è positiva se $\boldsymbol{x} \neq 0$ come nell'ipotesi.

cvd

Riassumendo, poniamo

$$\alpha = \begin{cases} \pm|\boldsymbol{x}|_2\frac{x_1}{|x_1|} & \text{se} x_1 \neq 0 \\ -|\boldsymbol{x}|_2 & \text{se} x_1 = 0 \end{cases}$$

$$\boldsymbol{v} = \boldsymbol{x} - \alpha\boldsymbol{e}_1$$

$\boldsymbol{e}_1$ modifica solo la prima componente del vettore $\boldsymbol{v}$. Quindi:

$$\begin{aligned} v_1 = x_1 - \alpha, &= x_1 - (-|\boldsymbol{x}|_2\frac{x_1}{|x_1|}) = \\ &= x_1 + |\boldsymbol{x}|_2\frac{x_1}{|x_1|}) = \\ &= x_1 * (\frac{|x_1| + |\boldsymbol{x}|_2}{|x_1|}) \end{aligned}$$

e non c'è possibilità di errori di cancellazione.

Se $\boldsymbol{x} \in \mathbb{R}^n$, $\alpha \in \mathbb{R}, \boldsymbol{v} \in \mathbb{R}^n$ per costruzione, allora P è una simmetrica ortogonale.

Il calcolo di $\boldsymbol{v}$ costa n operazioni per il calcolo della norma di $\boldsymbol{x}$, 2 operazioni moltiplicative, un'estrazione di radice, cioè un numero dell'ordine di n operazioni.

2.7.6 Algoritmo della fattorizzazione QR

Utilizziamo il teorema dimostrato per determinare la fattorizzazione QR.

Supponiamo di essere al passo k-esimo della fattorizzazione QR. Allora chiamiamo $\boldsymbol{a}_k = (A_{kk}, \dots, A_{nk})$ la k-esima sottocolonna della matrice $A^{(k)}$, che dobbiamo azzerare (corrisponde al vettore $\boldsymbol{x}$ del teorema). Scegliamo il vettore $\boldsymbol{v}_k = \boldsymbol{v}$ che ha 0 nelle prime $k-1$ componenti, e le componenti successive sono:

$$\boldsymbol{v}_k = \gamma_k(|\boldsymbol{a}_k|_2 + |A_{kk}^{(k)}|) = \mathrm{sgn} a_{kk} * (|\sigma_k|)$$

$$\boldsymbol{v}_{k+1} = a_{k+1}^{(k)}$$

$$\boldsymbol{v}_{k+2} = a_{k+2}^{(k)}$$

$$\dots$$

$$\boldsymbol{v}_n = a_n^{(k)}$$

dove definiamo

$$\gamma_k = \begin{cases} \frac{A_{kk}}{|A_{kk}|} & \iff A_{kk} \neq 0 \\ 1 & \mathrm{se} A_{kk} = 0 \end{cases}$$

(per la definizione di $\boldsymbol{v}_k$ sfruttiamo l'espressione ricavata nel teorema, cioè $\boldsymbol{v}_k = \boldsymbol{x} - \alpha \boldsymbol{e}_1$, e sfruttiamo anche l'espressione ricavata per v_1)

$$\beta_k = \frac{2}{(|\boldsymbol{v}_k|_2)^2}$$

Allora calcoliamo:

$$|\boldsymbol{v}_k|^2 = |\gamma_k|^2(|\boldsymbol{a}_k|_2 + |A_{kk}|)^2 + |\boldsymbol{a}_k|_2^2 - |A_{kk}|^2$$

$$= (|\boldsymbol{a}_k|_2)^2 + 2|A_{kk}||\boldsymbol{a}_k|_2 + (|A_{kk}|)^2 + |\boldsymbol{a}_k|_2^2 - |A_{kk}|^2$$

$$= 2(|\boldsymbol{a}_k|_2)^2 + 2|A_{kk}||\boldsymbol{a}_k|_2$$

e raccogliendo $2|\boldsymbol{a}_k|_2$ otteniamo:

$$= 2|\boldsymbol{a}_k|_2 * [|\boldsymbol{a}_k|_2 + |A_{kk}|]$$

cioè

$$\beta_k = \frac{2}{2(|\boldsymbol{v}_k|_2)^2}$$

$$= \frac{2}{2|\boldsymbol{a}_k|_2 * [|\boldsymbol{a}_k| + A_{kk}]}$$

$$\beta_k = \frac{1}{|\boldsymbol{a}_k|_2 * [|\boldsymbol{a}_k| + A_{kk}]}$$

Allora abbiamo determinato univocamente la matrice

$$P_k = I - \beta \boldsymbol{v}\boldsymbol{v}^h$$

Se il vettore $\boldsymbol{a}_k$ è tutto nullo, consideriamo $\gamma_k = 1$ e $\boldsymbol{v}_k = 0$, in modo che a quel passo la matrice di Hausolder sia l'identità.

Applicando la matrice P trovata al vettore $\boldsymbol{x}$ cioè alla colonna $\boldsymbol{a}_k$, tutte le entrate di $\boldsymbol{a}_k$ da $k+1$ a n vengono azzerate, mentre si ha $v_k = A_{kk} = -|\boldsymbol{x}|_2$ (infatti $P\boldsymbol{x} = \alpha \boldsymbol{e}_1$). Applichiamo poi la matrice di Hausolder alle colonne rimanenti di A.

Riepilogando, data la matrice A di partenza, con il primo passo $P_1 A$ azzero la prima colonna e modifico A_{11}, e poi ripeto il procedimento, per $n-1$ passi, e otteniamo la matrice triangolare superiore R.

Il prodotto di matrici unitarie è unitario, e sappiamo che

$$R = P_{n-1} \dots P_2 P_1 A$$

e per hermitianità:

$$A = P_1 P_2 \dots P_{n-1} R$$

quindi

$$Q = P_1 P_2 \dots P_{n-1}$$

2.7.7 Non unicità della fattorizzazione

Proposizione 2.3

Per ogni A esiste la fattorizzazione $A = QR$, ma non è unica.

Dim. Consideriamo una matrice di fase S, cioè una matrice diagonale con elementi $\theta_1, \dots, \theta_n$ con $|\theta_i| = 1 \forall i$, ed è una matrice unitaria. Allora $SS_h = I$, cioè $A = QR = QSS_hR$, e definiamo $\bar{Q}, \bar{R}$ dove $\bar{Q} = QS$ e $\bar{R} = S_h R$, dove $\bar{Q}$ è ancora unitaria e $\bar{R}$ è triangolare superiore. Allora abbiamo comunque una fattorizzazione QR della matrice.

cvd

Questa proprietà può essere sfruttata per scegliere la $\bar{R}$ in modo che gli elementi sulla diagonale siano ordinati in una certa maniera.

2.7.8 Fattorizzazione e risoluzione dei sistemi

Per risolvere il sistema lineare, non è necessario trovare esplicitamente la matrice Q, e nemmeno le P_h.

Passo 1: poniamo $A_1 = A, \boldsymbol{b}_1 = \boldsymbol{b}$, e chiamiamo T_1 la matrice che accosta $\boldsymbol{b}_1$.

Passo 2: chiamiamo $T_2 = P_1 T_1$, con

$$P_1 = I - \beta_1 \boldsymbol{v}_1 \boldsymbol{v}_1^h$$

Chiamiamo poi $\boldsymbol{y}_1^h$ il vettore riga $\boldsymbol{v}_1^h T_1$, in modo da scrivere:

$$T_2 = P_1 T_1 = T_1 - \beta_1 \boldsymbol{v}_1 \boldsymbol{y}_1^h$$

Passo $k+1$:

$$T_{k+1} = P_k T_k = T_k - \beta_k \boldsymbol{v}_k \boldsymbol{y}_k^h$$

con

$$\boldsymbol{y}_k^h = \boldsymbol{v}_k^h T_k$$

Infine
Passo n:

$$T_n = R \longrightarrow \boldsymbol{y}_k^h = \boldsymbol{v}_k R$$

il vettore $\boldsymbol{y}$ è stato calcolato senza calcolare esplicitamente le matrici Q, ed è tale che $q\boldsymbol{y} = \boldsymbol{b}$.

2.7.9 Costo computazionale

Per analizzare il costo del k-esimo passo, tengo conto che $\boldsymbol{v}_k$ ha $k-1$ componenti nulle e $n-k+1$ componenti diverse da 0. Per calcolare $\boldsymbol{v}_k$, le operazioni sono quindi $n-k+1+2 = n-k+3$.

.....

2.7.10 Stabilità della fattorizzazione QR

definizione|Si definisce **norma di FrobeniusTesto in grassetto** di A la quantità

$$\sqrt{\sum_{i,j=1}^{n} |A_{ij}|^2}$$

$= \sqrt{\mathrm{tr}(A_h * A)}$
Questa non è una norma matriciale indotta, infatti

$$|I|_f = \sqrt{n}$$

ed è diversa da 1, mentre con le norme matriciali indotte l'identità a norma 1.

Teorema 2.11 (analisi dell'errore all'indietro)

Sia $\bar{\boldsymbol{x}}$ la soluzione effettivamente calcolata con l'applicazione delle $\bar{P}_k$ effettive alla matrice A per calcolare la $\bar{R}$. Sia $\bar{\boldsymbol{y}}$ il vettore che compare nella risoluzione backward del sistema $R\boldsymbol{x} = \boldsymbol{y}$. Allora $\bar{\boldsymbol{x}}$ è soluzione esatta di un sistema

$$(A + \delta_A)\bar{\boldsymbol{x}} = \boldsymbol{b} + \delta_b$$

dove

$$|\delta_A|_f < u\gamma n^2 |A|_f + n|\bar{R}|_f + o(u^2)$$

$$|\delta_b|_2 < u\gamma n^2 |\boldsymbol{b}|_2 + O(u^2)$$

L'entità della perturbazione dipende dai dati iniziali, dalle dimensioni della matrice, e dalla matrice di fattorizzazione. Non dipende dalla norma di Frobenius di Q che è 1.

Consideriamo le matrici $A^{(k)}$ al k-esimo passo di fattorizzazione, e calcolo il prodotto

$$A_{k+1}^h A_{k+1} = A_k^h P_k^h P_k A_k$$

siccome $P_k^h P_k = I$ allora $A_k^h A_k$ non varia con i passi di fattorizzazione.

Allora

$$\mathrm{tr}(A_{k+1}^h A_{k+1}) = \mathrm{tr}(A_k^h A_k)$$

$$\longrightarrow |A_{k+1}|_f^2 = |A_k|_f^2$$

allora

$$|A_k|_f = |A|_f$$

inoltre

$$|R|_f = |A_n|_f = |A|_f$$

allora non c'è possibilità di esplosione dei coefficienti.

La QR è stabile e può essere fatta senza pivot.

2.8 Metodi iterativi

2.8.1 Descrizione e osservazioni introduttive

Nei metodi iterativi si crea una successione di vettori $\boldsymbol{x}_k = \phi \boldsymbol{x}_{k-1} + \boldsymbol{c}$, avendo assegnato un vettore $\boldsymbol{x}_0$ iniziale, e una matrice ϕ. Il metodo viene iterato finché $\boldsymbol{x}_k$ è un'approssimazione sufficientemente buona della soluzione.

I metodi iterativi sono convenienti quando A è una matrice sparsa, cioè quando il numero degli elementi diversi da 0 è trascurabile.

Applicando un metodo diretto a matrici di questo tipo, ho il fenomeno del *fill in*.

I metodi iterativi non alterano la struttura della matrice, e la loro operazione di base è quella di un prodotto matrice per vettore, che costa $o(n^2)$ operazioni se la matrice è piena, e $o(n)$ se la matrice è sparsa (vantaggio computazionale).

Per sua natura un metodo iterativo è meno sensibile all'errore algoritmico di un metodo diretto. Supponiamo che le operazioni iterative convergano. A causa dell'errore algoritmico la $\overline{\boldsymbol{x}}_*$ può variare, ma questo non importa perché con questi metodi non si cerca di calcolare una soluzione esatta, ma solo un'approssimazione.

Il fatto di troncare la successione causa la presenza di un errore analitico, che era assente nei metodi diretti.

Si dice che $\boldsymbol{x}_*$ converge a $\boldsymbol{x}_k$ se questo vale puntualmente, cioè se

$$\boldsymbol{x}_{*,i} = \lim_{k\to\infty} \boldsymbol{x}_{k,i}$$

Esempio 2.9 (metodo di Jacobi)

Consideriamo la i-esima equazione di un sistema $A\boldsymbol{x} = \boldsymbol{b}$, cioè

$$\sum_{j=1}^{n} A_{ij} x_j = b_i$$

ed esplicitando rispetto a x_i otteniamo

$$x_i = \frac{b_i - \sum_{i \neq j} A_{ij} x_j}{a_{ii}}$$

e al passo successivo otteniamo:

$$x_{k+1,i} = \frac{b_i - \sum_{i \neq j} A_{ij} x_j^{(k)}}{A_{ii}}$$

In forma compatta, questo metodo si scrive come:

$$\boldsymbol{x}_{k+1} = D^{-1}(\boldsymbol{b} - (L + U)\boldsymbol{x}_k)$$

dove

$$D = \mathrm{diag}(A_1)$$

$$L = \mathrm{tril}(A_1) \quad \text{triangolare dalla prima sottodiagonale a scendere}$$

$$U = \mathrm{triu}(A_1)$$

Si ha quindi una scrittura della forma

$$\boldsymbol{x}_{k+1} = \phi \boldsymbol{x}_k + \boldsymbol{c}$$

2.8.2 Generalità sulle tecniche di splitting

Dato il sistema lineare $A\boldsymbol{x} = \boldsymbol{b}$, $A \in \mathbb{C}_{n,n}$, chiamiamo $\boldsymbol{x}_*$ la soluzione esatta. Consideriamo uno splitting della matrice A che viene scritta come $M - N$, dove M, N sono matrici quadrate e M è invertibile.

Applichiamo lo *splitting* alla matrice A nel sistema lineare, cioè

$$A\boldsymbol{x} = M\boldsymbol{x} - N\boldsymbol{x} = \boldsymbol{b}$$

e posso riscrivere

$$\boldsymbol{x} = M^{-1}N\boldsymbol{x} + M^{-1}\boldsymbol{b}$$

allora poniamo $P = M^{-1}N$ e $\boldsymbol{c} = M^{-1}\boldsymbol{b}$ quindi abbiamo il sistema

$$\boldsymbol{x}_k = P\boldsymbol{x} + \boldsymbol{c}$$

Il metodo iterativo è

$$\boldsymbol{x}_{k+1} = P\boldsymbol{x}_k + \boldsymbol{c} \quad \text{relazione} \triangle$$

e P viene chiamata matrice di iterazione.

Supponiamo che la successione dei vetori $\boldsymbol{x}_k$ converga, allora

$$\lim_{k \to \infty} \boldsymbol{x}_k = \boldsymbol{x}_*$$

Passando al limite nella relazione $\triangle$:

$$\boldsymbol{x}_* = P\boldsymbol{x}_* + \boldsymbol{c}, \quad \text{relazione} \circ$$

Sottraendo membro a membro le relazioni $\triangle$ e $\circ$ otteniamo:

$$\boldsymbol{x}_* - \boldsymbol{x}_{k+1} = P\boldsymbol{x}_* + \boldsymbol{c} - (P\boldsymbol{x}_k + \boldsymbol{c}) = P(\boldsymbol{x}_* - \boldsymbol{x}_k)$$

dove $\boldsymbol{x}_* - \boldsymbol{x}_{k+1}$ è l'errore assoluto E_{k+1} al passo $k+1$-esimo, e si ha

$$E_{k+1} = PE_k = P^2 E_{k-1} = \cdots = P^{k+1} E_0$$

Per ogni k,

$$E_k = P^k E_0$$

2.8.3 Convergenza del metodo iterativo

$\Diamond$ **Definizione** Diciamo che un **metodo iterativo** è **convergente** se

$$\lim_{k\to\infty} \boldsymbol{x}_k = \boldsymbol{x}_*$$

per ogni vettore d'innesco $\boldsymbol{x}_0$.

Il metodo iterativo $\triangle$ è convergente se e solo se

$$\lim_{k\to\infty} E_k = 0$$

per ogni E_0, e quindi se

$$\lim_{k\to\infty} P^k = 0$$

Teorema 2.12 (condizione necessaria e sufficiente per la convergenza)

Il metodo è convergente, cioè

$$\lim_{k\to\infty} P^k = \text{matrice nulla}$$

se e solo se $\rho(P) \leq 1$.

Dimostrazione

$1 \longrightarrow 2$: supponiamo che il metodo sia convergente, allora

$$\lim_{k\to\infty} E_k = 0$$

Scegliamo $\boldsymbol{x}_0$ in modo tale che $E_k = \boldsymbol{x}_* - \boldsymbol{x}_0$ sia un autovettore rispetto a λ che realizza il raggio spettrale, quindi

$$E_k = P^k E_0 = \lambda^k E_0$$

(infatti E_0 è autovettore)

$$E_k = |P^k E_0| = |\lambda|^k |E_0| = \rho(P)^k |E_0|$$

allora

$$\lim_{k\to\infty} \rho(P)^k = 0$$

$|\rho(P)| < 1$.

$2 \longrightarrow 1$:

Caso di P diagonalizzabile: P può essere scritta come $P = S\lambda S^{-1}$ dove λ è la matrice diagonale con entrate uguali agli autovalori (la maggior parte delle matrici sono diagonalizzabili).

Allora

$$P^k = (S\lambda S^{-1}) * (S\lambda S^{-1})(S\lambda S^{-1}) \dots (S\lambda S^{-1}) = S\lambda^k S^{-1}$$

e P^k tende a 0 se λ tende a 0, cioè se e solo se $|\lambda_i| \leq 1 \forall i$, quindi se e solo se $\rho(P) \leq 1$.

Caso generale: per dimostrare questo caso serve la nozione di forma normale di Shur.

Teorema 2.13 (forma normale di Shur)

Data una matrice $A \in \mathbb{C}_{n,n}$ con autovalori $\lambda_1, \dots, \lambda_n$, allora esiste Q unitaria e una matrice T triangolare superiore con elementi diagonali $t_{ii} = \lambda_i$ tali che $A = QTQ_h$.

Se A è hermitiana, T diventa la matrice λ degli autovalori.

Consideriamo la forma di Shur di P, cioè

$$P = QTQ_h$$

Allora consideriamo

$$P^k = (QTQ_h)(QTQ_h) \dots (QTQ_h)$$

e siccome Q è unitria, $QQ_h = I$, quindi rimane

$$P^k = QT^kQ_h$$

allora

$$\lim_{k\to\infty} P^k = 0$$

se e solo se

$$\lim_{k\to\infty} T^k = 0$$

Inoltre, $\rho(P) = \rho(T)$ perché le due matrici sono simili, e si ha

$$0 \leq |T^k| \leq |T|^k \leq (|T + E|)^k = |F|^k$$

Definisco la matrice E in modo che:

1. E sia una matrice diagonale, non negativa
2. $|F| = |T + E| \leq 1$,
3. gli elementi diagonali di $T + E$ sono a due a due distinti, cioè richiedo che la matrice $T + E$ sia diagonalizzabile.

Allora, per F diagonalizzabile, ho già dimostrato che

$$\lim_{k\to\infty} |F|^k = 0$$

se e solo se

$$\rho(|F|^k) \leq 1$$

Allora per confronto

$$\lim_{k\to\infty} T^k = 0$$

e segue la tesi.

cvd

Supponiamo di considerare $|T|$ e di ordinarne gli autovalori per modulo decrescente, in modo che $\tau_1 \geq \tau_2 \geq \tau_3$. Allora la matrice E è la matrice diagonale con entrate $\varepsilon_1, \ldots, \varepsilon_t$ dove

$$\varepsilon_1 = (1 - \tau_1)/2$$

$$\varepsilon_2 = \varepsilon_1/2$$

$$\varepsilon_t = \varepsilon_{t-1}/2$$

Con questa tecnica si riesce a garantire che gli autovalori di $T + E$ sono a due a due distinti.

2.8.4 Altre osservazioni sulla convergenza

Vale la seguente proposizione:

Proposizione 2.4 [condizione sufficiente di convergenza]

Se esiste una norma matriciale indotta tale $|P| \leq 1$ allora il metodo è convergente.

Dim. Per ogni norma matriciale indotta vale che $\rho(P) \leq |P|$, allora $\rho(P) \leq 1$.

cvd

Osservazione 2.6

Supponiamo che esista una norma tale che

$$|P| \leq c \leq 1$$

(cioè il metodo converge)

Allora, usando la stessa norma, l'errore decresce in modo monotono.

Dim. Consideriamo

$$|E_k| = |PE_{k-1}| \leq |P| * |E_{k-1}| \leq c * |E_{k-1}| \leq |E_{k-1}|$$

Inoltre sappiamo che

$$|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}| = |P * \boldsymbol{x}_{k-1} + \boldsymbol{c} - (P * \boldsymbol{x}_{k-2} + \boldsymbol{c})| =$$

$$= |P(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-2})| \leq |P| * |\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-2}|$$

$$\leq c * |\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-2}|$$

e si ha una decrescita monotona anche dell'incremento.

cvd

Tuttavia $\rho(P) \leq 1$ non implica che $|P| \leq 1$ per qualsiasi norma, e l'errore prima di convergere può avere una crescita iniziale.

Esempio 2.10

Consideriamo una matrice d'iterazione

$$\begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix}$$

$$\rho(P) = 0$$

invece $|P|_2 = 2$, infatti

$$P_h = \begin{pmatrix} 0 & 0 \\ 2 & 0 \end{pmatrix}$$

$$P_h P = \begin{pmatrix} 4 & 0 \\ 0 & 0 \end{pmatrix}$$

e il polinomio caratteristico è

$$p_\lambda = \lambda^2 - 4\lambda$$

quindi $\sqrt{\max |\rho(A_h A)|} = 2$.

$$|E_0|_2 = 0,\ |E_1|_2 = 2,\ |E_2|_2 = 0$$

Quindi, prima che il metodo converga, l'errore ha una crescita iniziale.

Stiamo considerando metodi che generano la successione

$$\boldsymbol{x}_{k+1} = P\boldsymbol{x}_k + \boldsymbol{c}$$

tali che $\rho(P) < 1$. In uno spazio finitodimensionale, non abbiamo la dipendenza dalla norma.

Infatti

$$T(\boldsymbol{x}) = P\boldsymbol{x} + \boldsymbol{c}$$

$T(\boldsymbol{x}) \to \boldsymbol{x}_*$ se e solo se $\rho(P) < 1$.

$\rho(P)$ è l'inf delle norme matriciali indotte della norma di P. Questo implica che, se $\rho(P) < 1$, allora esiste una norma matriciale indotta $*$ tale che

$$|P|_* \leq 1$$

quindi

$$|\boldsymbol{x} - \boldsymbol{y}| = |P\boldsymbol{x} + \boldsymbol{c} - (P\boldsymbol{y} + \boldsymbol{c})| = |P(\boldsymbol{x} - \boldsymbol{y})|_*$$

$$\leq |P|_* * |\boldsymbol{x} - \boldsymbol{y}|_*$$

allora per il teorema di Banach abbiamo convergenza.

Siccome tutte le norme in uno spazio finitodimensionale sono topologicamente equivalenti, si ha convergenza rispetto a una qualsiasi norma.

Se $|E_k|_* \to 0$, siccome $|E| = K^*|E_k|_*$, anche la norma a primo membro tende a 0.

2.9 Principali metodi iterativi

2.9.1 Metodo di Jacobi

Dividiamo A in tre matrici $D+L+U$, dove D è la matrice diagonale, L la triangolare inferiore stretta (con 0 sulla diagonale principale), U triangolare superiore stretta.

Nel caso del metodo di Jacobi $A = M - N$ con $M = D$ e $N = -(L+U)$ (N è uguale ad A con zeri sulla diagonale principale).

$$A\boldsymbol{x} = (M - N)\boldsymbol{x} = \boldsymbol{b}$$

$$\longrightarrow \boldsymbol{x} = M^{-1}N\boldsymbol{x} + M^{-1}\boldsymbol{b}$$

$$A = M^{-1}N, \quad \boldsymbol{c} = M^{-1}\boldsymbol{b}$$

e tenendo conto delle espressioni di M e N:

$$P = -D^{-1}(L+U),\ \boldsymbol{c} = D^{-1}\boldsymbol{b}$$

Indicando con P_j la matrice d'iterazione del metodo di Jacobi si ha:

$$\boldsymbol{x}_k = P_j\boldsymbol{x}_{k-1} + \boldsymbol{c}_j$$

e moltiplicando per D:

$$D\boldsymbol{x}_k = DP_j\boldsymbol{x}_{k-1} + D\boldsymbol{c}_j$$

e sostituendo le espressioni ricavate prima:

$$D\boldsymbol{x}_k = -(L+U)\boldsymbol{x}_{k-1} + \boldsymbol{b}$$

Considerando la i-esima equazione si ha:

$$x_i = \frac{-\sum_{j\neq i} A_{ij}x_j^{(k-1)} + b_i}{A_{ii}}$$

2.9.2 Metodo di Gauss-Seidel

In questo caso si pone $M = D + L$, invece $N = -U$, e si ricava:

$$P = -(D+L)^{-1}U, \quad \boldsymbol{c} = (D+L)^{-1}\boldsymbol{b}$$

Come prima, moltiplichiamo per M la relazione

$$\boldsymbol{x}_k = P\boldsymbol{x}_{k-1} + \boldsymbol{b}$$

e otteniamo

$$(D+L)\boldsymbol{x}_k = -U\boldsymbol{x}_{k-1} + \boldsymbol{b}$$

cioè

$$D\boldsymbol{x}_k = -L\boldsymbol{x}_k - U\boldsymbol{x}_{k-1} + \boldsymbol{b}$$

Considerando l'equazione i-esima,

$$x_i^{(k)} = \frac{-\sum_{j<i} A_{ij}x_j^{(k)} - \sum_{j>i} A_{ij}x_j^{(k-1)}}{A_{ii}}$$

Teniamo conto che nel prodotto con la triangolare inferiore si utilizzano le informazioni già aggiornate al passo k-esimo, e sopravvivono solo gli elementi con colonne di indice minori di i. Nell'altro prodotto invece si utilizzano le informazioni ricavate al passo $k-1$.

2.9.3 Matrici diagonalmente dominanti e irriducibili

◊ **Definizione** Data una matrice $A \in \mathbb{C}_{n,n}$ diciamo che:

1. A è diagonalmente dominante in senso stretto per righe se vale

$$|A_{ii}| > \sum_{j \neq i} |A_{ij}| \quad i = 1, \ldots, n$$

2. A è diagonalmente dominante in senso stretto per colonne se vale

$$|A_{ii}| > \sum_{i \neq j} |A_{ji}|$$

3. A è diagonalmente dominante in senso debole per righe se

$$|A_{ii}| \geq \sum_{j \neq i} |A_{ij}|$$

e se esiste almeno un indice $\bar{i}$ tale che

$$|A_{\bar{i},\bar{i}}| > \sum_{j=1}^{n} |A_{\bar{i},j}|$$

◊ **Definizione** $A \in \mathbb{C}_{n,n}$ si dice riducibile se esiste π matrice di permutazione tale che

$$B = \pi A \pi^t$$

può essere descritta a blocchi come:

$$\begin{matrix} A_{11} & A_{12} \\ 0 & A_{22} \end{matrix}$$

con il primo blocco fatto da k righe e l'ultimo da $n-k$ righe.

Se A, B sono come sopra, dato il sistema lineare

$$B\boldsymbol{x} = \boldsymbol{b}$$

esso si può riscrivere come:

$$\begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

allora si ottengono le equazioni

$$\begin{cases} A_{11}x_1 + A_{12}x_2 = b_1 \\ A_{22}x_2 = b_2 \end{cases}$$

e risolvendo prima la seconda e sostituendo poi x_2 nella prima otteniamo un sistema disaccoppiato.

Esempio 2.11

Consideriamo le matrici

$$\begin{matrix} 1 & 3 & 0 \\ 0 & 2 & -1 \\ 0 & 0 & 2 \end{matrix}$$

Associamo questa matrice di ordine $n = 3$ a un grafo con tre nodi $1, 2, 3$. Per ogni entrata della matrice, quando $A_{ij} \neq 0$ collego il nodo i con il nodo j con un arco orientato da i a j. Ad esempio, $A_{ii} = 1$ quindi traccio un arco che entra ed esce nel nodo 1. Nel caso di questa matrice, possiamo andare dal nodo 1 al 2 seguendo l'arco orientato tracciato in corrispondenza di $A_{12} = 3$, possiamo andare dal nodo 2 al 3 lungo l'arco tracciato in corrispondenza di $A_{23} = -1$, ma non possiamo andare in nessun modo dal nodo 3 all'1 perché $A_{31} = 0$.

Consideriamo invece la matrice

$$\begin{matrix} 1 & 3 & 0 \\ 0 & 2 & -1 \\ -1 & 0 & 2 \end{matrix}$$

Nel caso della seconda matrice, c'è un cammino chiuso che connette tutti i nodi.

Teorema 2.14

Una matrice A è irriducibile se e solo se il grafo associato è fortemente connesso ovvero se e solo se esiste un cammino dal nodo i al nodo j per ogni coppia di nodi i, j.

Quindi la prima matrice è riducibile e la seconda non lo è.

2.9.4 Condizioni sufficienti di convergenza

Le condizioni sufficienti per la convergenza del metodo di Jacobi sono le seguenti:

1. A è diagonalmente dominante in senso stretto (per righe o colonne)

Dim. Sappiamo che

$$|A|^{\infty} = \max_{i=1}^{n}[\sum_{j=1}^{n} |A_{ij}|]$$

Nel metodo di Jacobi la matrice di permutazione è

$$P = M^{-1}N = -D^{-1}(L + U)$$

Se calcolo la norma infinito della matrice di Jacobi P_j, si ha:

$$|P| \leq |D^{-1}| * |L + U|$$

Sfruttando il fatto che A è diagonalmente dominante in senso stretto sulle righe, si ha:

$$|L + U| \leq \max_i |A_{ii}|$$

(infatti la somma di ogni riga privata dell'elemento diagonale è minore dell'elemento diagonale)

$$|D^{-1}| = \frac{1}{\max_i |A_{ii}|}$$

quindi

$$|P| \leq \frac{1}{\max_i |A_{ii}|} * \max_i |A_{ii}| \leq 1$$

$\rho(P) < 1$ e il metodo converge.

2. A è diagonalmente dominante in senso debole (per righe o per colonne)
3. A è irriducibile.

Nelle stesse ipotesi converge anche il metodo di Gauss-Seidel, però per questo metodo vale anche la seguente proposizione:

Proposizione 2.5

Data una matrice A $m \times n$ hermitiana, non singolare il metodo di Gauss-Seidel converge se e solo se A è definita positiva (la fattorizzazione di Cholesky, che determina se una matrice è definita positiva oppure no, è un buon modo per determinare se il metodo di Gauss-Seidel converge o no).

Teorema 2.15

Sia $A \in \mathbb{C}_{n,n}$ tridiagonale, siano

$$\boldsymbol{a} = (a_1, \ldots, a_n) \text{ diagonale principale}$$

$$\boldsymbol{c} = (c_1, \ldots, c_{n-1}) \text{ sopradiagonale}$$

$$\boldsymbol{b} = (b_2, \ldots, b_n) \text{ sottodiagonale}$$

Se λ è un autovalore della matrice di iterazione di Jacobi, allora λ^2 è un autovalore per P_g (matrice di iterazione di Gauss-Seidel). Inoltre, se $\lambda \neq 0$ è un autovalore di P_g, allora $\pm\sqrt{\lambda}$ è un autovalore di P_j.

Allora per le matrici tridiagonali, $\rho(P_g) = \rho(P_j)^2$, allora Gauss-Seidel è convergente se e solo se Jacobi è convergente. Inoltre se Gauss-Seidel converge, ha un tasso di convergenza del doppio rispetto a Jacobi, infatti il tasso asintotico di convergenza è $\log \rho(P)$ e si ha

$$\frac{\rho(P_g)}{\rho(P_j)} = \frac{\log \rho(P_j)}{\log \rho(P_j)^2} = 1/2$$

(tuttavia non è detto che sia sempre più conveniente usare Gauss-Seidel).

2.9.5 Tasso di convergenza

Consideriamo la relazione:

$$|E_k| \leq |P^k| * |E_{k-1}|$$

Allora valgono le seguenti definizioni

fattore di convergenza	$= \lvert P^k \rvert$
fattore di convergenza medio	$= \lvert P^k \rvert^{1/k}$
tasso di convergenza medio	$R_k(P) = -1/k \log \lvert P^k \rvert$
tasso asintotico di convergenza	$R(P) = \lim_{k\to\infty} r_k(P) = \log \rho(P)$

La definizione di tasso asintotico di convergenza non dipende né dalla norma né dal passo.

2.9.6 Test d'arresto test dell'incremento

Si considera la differenza fra un'iterata e la successiva, misurata con una certa norma, e si richiede che il metodo si arresti quando è verificata la condizione

$$|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}| \leq \varepsilon$$

dove ε è una tolleranza assegnata.

$$E_k = |\boldsymbol{x}_* - \boldsymbol{x}_k|$$

$$\longrightarrow |\boldsymbol{x}_k - \boldsymbol{x}_{k-1}| = |\boldsymbol{x}_k - \boldsymbol{x}_*| + |\boldsymbol{x}_* - \boldsymbol{x}_{k-1}| \leq E_k + E_{k-1}$$

$$= E_{k-1} + P * E_{k-1} = (I + P) * E_{k-1}$$

Quindi

$$\boldsymbol{x}_k - \boldsymbol{x}_{k-1} \leq (I + P)E_{k-1}$$

$$\longrightarrow E_{k-1} = (I + P)^{-1}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$

Passando alla norma:

$$|E_{k-1}| \leq |(I + P)^{-1}| * |\boldsymbol{x}_k - \boldsymbol{x}_{k-1}| \quad \text{relazione } *$$

Applichiamo il teorema: se $|A| < 1$ con norma matriciale indotta, allora $I + A$ è non singolare e

$$|(I + A)^{-1}| \leq \frac{1}{1 - |A|}$$

Allora, sostituendo l'espressione trovata per $|(I + P)^{-1}|$ nella relazione $*$ ottengo:

$$|E_{k-1}| \leq \frac{1}{1 - |P|} * |\boldsymbol{x}_k - \boldsymbol{x}_{k-1}|$$

Quindi

$$|E_k| = |PE_{k-1}| \leq |P| * |E_{k-1}|$$

$$= \frac{|P|}{1 - |P|}|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}|$$

Il test è affidabile tranne nel caso in cui $|P| \simeq 1$.

2.9.7 Test d'arresto Test del residuo

Il residuo è la quantità $\boldsymbol{b} - A\boldsymbol{x}$. Si definisce il residuo al passo k come $\boldsymbol{b} - A\boldsymbol{x}_k$. Scelta una norma, si impone che la norma del residuo sia minore di ε, e il metodo si arresta quando questa condizione è verificata.

Sapendo che

$$A\boldsymbol{x}_* = \boldsymbol{b}$$

allora

$$R_k = \boldsymbol{b} - A\boldsymbol{x}_k = A * (\boldsymbol{x}_* - \boldsymbol{x}_k) = A * E_k \quad \text{equazione dell'errore}$$

allora

$$E_k = A^{-1}R_k$$

$$|E_k| = |A^{-1}R_k| \leq |A^{-1}| * |R_k|$$

$$= \frac{K(A)}{|A|} * |R_k|$$

Allora l'errore assoluto è controllato bene dal residuo, se il numero di condizionamento non è molto grande.

Il test del residuo è affidabile se il problema non è mal condizionato.

Se invece vogliamo stimare l'errore relativo:

$$\frac{|E_k|}{|\boldsymbol{x}_*|}$$

$$\leq \frac{|A^{-1}| * |R_k|}{|\boldsymbol{x}_*|}$$

Allora, osserviamo che:

$$|\boldsymbol{b}| \leq |A| * |\boldsymbol{x}_*|$$

$$\longrightarrow \frac{1}{|\boldsymbol{x}_*|} \leq \frac{|A|}{|\boldsymbol{b}|}$$

allora

$$\frac{|E_k|}{|\boldsymbol{x}_*|} \leq \frac{|A^{-1}| * |A| * |R_k|}{|\boldsymbol{b}|}$$

$$= K(A) * \frac{|R_k|}{|\boldsymbol{b}|}$$

La quantità $\frac{|R_k|}{|\boldsymbol{b}|}$ viene chiamata residuo relativo, se supponiamo $r_0 = \boldsymbol{b}$ se $\boldsymbol{x}_0 = 0$. L'errore relativo viene controllato dal residuo relativo a meno di condizionamento.

Il costo computazionale dell'iterazione è $o(n^2)$ operazioni per iterazione, mentre se la matrice è sparsa il costo è $o(n)$.

2.10 Metodi di rilassamento

2.10.1 Definizione del metodo

Definiamo il metodo di Gauss-Seidel come:

$$\boldsymbol{x}_k = D^{-1}(\boldsymbol{b} - L\boldsymbol{x}_k - U\boldsymbol{x}_{k-1})$$

Possiamo scrivere $\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \boldsymbol{v}_k$ con $\boldsymbol{v}_k = \boldsymbol{x}_k - \boldsymbol{x}_{k-1}$. La nuova iterata è il punto in $\mathbb{C}_n$ che otteniamo a partire da $\boldsymbol{x}_{k-1}$ facendo un passo di lunghezza $|\boldsymbol{v}_k|$ nella direzione di $\boldsymbol{v}_k$.

Viene definita una nuova classe di metodi, detti di rilassamento, dove

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \omega * \boldsymbol{v}_k$$

dove si fa un passo di lunghezza $\omega * |\boldsymbol{v}_k|$.

Se $\omega < 1$ abbiamo un sottorilassamento, mentre se $\omega > 1$ abbiamo un soprarilassamento.

Un metodo di questo tipo viene chiamato SOR.

Sostituiamo nella relazione che definisce il metodo SOR l'espressione di $\boldsymbol{v}_k$ che deriva da Gauss-Seidel.

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \omega * (\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) = (1-\omega)\boldsymbol{x}_{k-1} + \omega \boldsymbol{x}_k$$

$$= (1-\omega)\boldsymbol{x}_{k-1} + \omega(D^{-1}(\boldsymbol{b} - L\boldsymbol{x}_k - U\boldsymbol{x}_{k-1}))$$

e mettendo in evidenza D^{-1} si ha:

$$\boldsymbol{x}_k = D^{-1}[(1-\omega)D * \boldsymbol{x}_{k-1} + \omega \boldsymbol{b} - \omega L\boldsymbol{x}_k - \omega U\boldsymbol{x}_{k-1}]$$

Raccogliendo tutti i termini che moltiplicano $\boldsymbol{x}_k$:

$$(I + \omega D^{-1}L)\boldsymbol{x}_k = D^{-1}((1-\omega)D - \omega U)\boldsymbol{x}_{k-1} + \omega D^{-1}\boldsymbol{b}$$

Moltiplicando per D:

$$(D + \omega L)\boldsymbol{x}_k = ((1-\omega)D - \omega U)\boldsymbol{x}_{k-1} + \omega \boldsymbol{b}$$

Allora

$$P = (D + \omega L)^{-1} * ((1-\omega)D - \omega U)$$

è la matrice di iterazione, invece

$$\boldsymbol{c} = \omega(D + \omega L)^{-1}\boldsymbol{b}$$

Allora

$$D\boldsymbol{x}_k = ((1-\omega)D - \omega U)\boldsymbol{x}_{k-1} - \omega L\boldsymbol{x}_k + \omega \boldsymbol{b}$$

Considerando la i-esima componente:

$$x_i^{(k)} = (1-\omega)x_i^{(k-1)} + \frac{\omega}{A_{ii} * [b_i - \sum_j < i} a_{ij}x_j^{(k)} - \sum j > i a_{ij}x_j^{(k+1)}]$$

e per $\omega = 1$ si ha il metodo di Gauss-Seidel.

Il costo computazionale di questo metodo è vicino a quello di Gauss-Seidel.

2.10.2 Ottimizzazione del metodo

Si vuole determinare l'ω ottimale, tale per cui il metodo converge più velocemente.

Teorema 2.16 (teorema di Kan)

Condizione necessaria per la convergenza del SOR è che $|\omega - 1| < 1$, ovvero $\omega \in (0, 2)$ se ω è reale.

Vale la seguente condizione sufficiente per la convergenza:

Teorema 2.17 (Teorema di Ostrosky-Right)

Se A è definita positiva e $\omega \in (0, 2)$ allora il metodo SOR è convergente.

Dimostrazione

La matrice di iterazione è

$$P_\omega = (D + \omega L)^{-1} * ((1-\omega)D - \omega U)$$

$D+\omega L$ è una matrice triangolare inferiore con elementi $a_{11}, \dots a_{nn}$ sulla diagonale principale, e il determinante è uguale al prodotto degli elementi sulla diagonale, cioè è uguale a $\det D$.

La matrice $(1-\omega)D - \omega U$ è una triangolare superiore con elementi diagonali $(1-\omega)a_{11}, \dots, (1-\omega)a_{nn}$ mentre la parte triangolare ha entrate ωa_{ij}. Il determinante è ancora dato dal prodotto degli elementi sulla diagonale principale, e quindi è $(1-\omega)^n \det D$.

Per il teorema di Binet:

$$\det P_\omega = \det[(1-\omega)D - \omega U] * \frac{1}{\det(D+\omega L)} = \frac{(1-\omega)^n * \det D}{\det D} = (1-\omega)^n$$

Inoltre

$$\det P_\omega = \prod_{i=1}^{n} \lambda_i(P_\omega) =$$

Siccome ciascun autovalore di P_ω è minore del raggio spettrale, si ha

$$|1-\omega|^n \leq \rho^n(P_\omega)$$

e se deve esserci convergenza dev'essere

$$\rho(P_\omega) \leq 1$$

allora la relazione implica $|1-\omega| \leq 1$.

cvd

Nel caso delle matrici tridiagonali vale il seguente teorema:

Teorema 2.18

Sia $A \in \mathbb{C}_{n,n}$ tridiagonale. Se P_j (matrice d'iterazione di Jacobi) è tale che $\rho(P_j) < 1$, e ha autovalori reali, allora esiste ed è unico ω^* tale che

$$\rho(P_\omega^*) = \min_{\omega \in (0,2)} \{\rho(P_\omega)\}$$

Sappiamo che $\omega^* = \frac{2}{1+\sqrt{1-\rho^2(P_j)}}$.

È possibile rappresentare la variazione del raggio spettrale in funzione di ω: traccio un grafico con ω in ascissa e p_ω in ordinata, allora il grafico scende, ha una cuspide in ω^* e poi risale in maniera lineare.

Per $\omega > \omega^*$, $\rho(P(\omega^*)) = \omega - 1$.

Conviene stare a destra di ω^*, perché in questo modo il raggio spettrale sarà più grande di poco, invece a sinistra si è in prossimità della cuspide e il valore varia di molto. Conviene approssimare per eccesso.

3

Localizzazione degli autovalori

3.1 Localizzazione degli autovalori

3.1.1 Cerchi di Gerschgorin

◊ **Definizione** Sia $A \in \mathrm{Mat}_{n\times n,\mathbb{C}}$ e definiamo i-esimo cerchio di Gerschgorin per righe l'insieme

$$K_i = \{z \in \mathbb{C}\, t.c.\, |z - A_{ii}| \leq \sum_{j\neq i} |A_{ij}|\}$$

(sono cerchi centrati nell'entrata diagonale i-esima e con raggio minore della somma delle entrate fuori diagonale della stessa riga). Analogamente si definiscono i cerchi di Gerschgorin per colonna:

$$\{z \in \mathbb{C}\, t.c.\, |z - A_{jj}| \leq \sum_{i\neq j} |A_{ij}|\}$$

3.1.2 I tre teoremi di localizzazione

Teorema 3.1

Lo spettro σ_A della matrice A, cioè l'insieme dei suoi autovalori, è contenuto nell'unione dei cerchi di Gerschgorin, cioè è contenuta in

$$\bigcup_{i=1}^{n} K_i$$

Teorema 3.2

Sia $M_1 = \bigcup_{i=i_1,\ldots,i_k} K_i$ (unione di k cerchi) e $M_2 = \bigcup_{i=i_{k+1},\ldots,i_n} K_i$ (unione di $n-k$ cerchi) dove $M_1 \cap M_2 = \emptyset$. Allora k autovalori appartengono a M_1 e $n-k$ autovalori appartengono a M_2.

Esempio 3.1

Consideriamo $A \in M_{4,\mathbb{R}}$, allora se c'è un autovalore complesso anche il suo coniugato è un autovalore di A, e dev'essere contenuto nello stesso cerchio K_i. Supponiamo che i cerchi di Gerschgorin relativi ad A sono tali che K_1 è disgiunto da $K_2 \cup K_3 \cup K_4$. Allora, siccome per il teorema 2 nel cerchio K_1 c'è solo un autovalore, si può dire che la matrice ha almeno due autovalori reali. Infatti l'autovalore contenuto in C_1 è necessariamente reale, perché C_1 non contiene il coniugato di questo autovalore. L'unione degli altri tre cerchi contiene tre autovalori, allora almeno uno di questi sarà sicuramente reale.

Teorema 3.3

Sia A una matrice irriducibile. Se l'autovalore λ_A appartiene alla frontiera dell'unione di tutti i cerchi, allora esso appartiene alla frontiera di ogni cerchio.

Equivalentemente, il terzo teorema può essere riscritto nel seguente modo: supponiamo di avere $\lambda \in \partial \bigcup_i K_i$, e supponiamo di avere un indice j tale che $\lambda \notin \partial K_j$, allora λ non è autovalore.

Esempio 3.2

Consideriamo la matrice con 2 sulla diagonale principale e con -1 sulla sopradiagonale e sottodiagonale. Alla prima e all'ultima riga corrispondono cerchi di raggio 1 e centro 2. Invece i cerchi corrispondenti alle righe dalla seconda alla penultima sono di raggio 2 e centro 2.

Allora per il teorema 3, A è non singolare (0 non è autovalore), infatti 0 non appartiene alla frontiera dei cerchi di raggio 1.

3.1.3 Dimostrazione del primo teorema di localizzazione

Sia $A\boldsymbol{x} = \lambda \boldsymbol{x}$, $\boldsymbol{x} \neq 0$. Consideriamo la i-esima equazione:

$$\sum_{j=1}^{n} A_{ij} x_j = \lambda x_i$$

e separando le entrate A_{ii} otteniamo:

$$(\lambda - A_{ii}) x_i = \sum_{j \neq i} A_{ij} x_j \quad \text{relazione } *$$

Consideriamo la norma infinito di $\boldsymbol{x}$, allora esiste p tale che $|\boldsymbol{x}|^\infty = x_p$.

Allora riscrivo la relazione $*$ come:

$$|(\lambda - A_{pp}) x_p| = |\sum_{j \neq p} A_{pj} x_j|$$

$$|x_j| \leq |x_p| \longrightarrow \quad |\lambda - a_{pp}| * |x_p| \leq \sum_{j \neq p} A_{pj} x_p|$$

ed elidendo x_p:

$$|\lambda - A_{pp}| \leq \sum_{j \neq p} |A_{pj}|$$

allora λ appartiene al cerchio K_p. p non è noto, allora λ appartiene all'unione di tutti i cerchi.

cvd

Osservazione 3.1

Supponiamo di avere una matrice A diagonalmente dominante in senso stretto, oppure che sia diagonalmente dominante in senso debole e irriducibile. Allora A è non singolare. Se inoltre A è hermitiana e gli A_{ii} sono reali positivi, allora A è definita positiva.

Dim. Siccome A è diagonalmente dominante

$$|A_{ii}| > \sum_{j \neq i} |A_{ij}|$$

quindi la distanza dell'autovalore dall'origine è maggiore del raggio, e i cerchi sono lontani dall'origine. Quindi 0 non appartiene alla loro frontiera.

Nel caso di dominanza diagonale debole e irriducibilità, abbiamo almeno un cerchio che tocca l'origine, perché esiste i tale che

$$|A_i| = r_i$$

allora $0 \in \partial K_i$. Per l'irriducibilità esiste un altro indice s tale che $A_{ss} > r_s$, allora $0 \notin \partial K_s$. Allora per il terzo teorema 0 non può essere autovalore.

Se gli elementi diagonali della matrice sono positivi, i cerchi sono tutti nel semipiano positivo, gli autovalori sono positivi e la matrice è definita positiva.

cvd

3.2 Calcolo di autovalori estremi

3.2.1 Osservazione 3.2

Gli autovalori di una matrice sono le radici del polinomio caratteristico, ma non è conveniente calcolarli in questo modo. Al contrario, a volte il problema del calcolo delle radici di un polinomio viene ricondotto al calcolo degli autovalori di una matrice. Infatti dato un polinomio di grado n della forma

$$p_n = \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0,$$

associo a esso una particolare matrice della forma:

$$\begin{matrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & a_{n-1} \\ 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & a_1 \\ 0 & 0 & 0 & 1 & a_0 \end{matrix}$$

3.2.2 Condizionamento

Teorema 3.4 (teorema di Bauer-Flick)

Sia A una matrice in $\mathrm{Mat}_{n,\mathbb{C}}$ diagonalizzabile, della forma $S\lambda S^{-1}$. Sia $E \in \mathrm{Mat}_{n,\mathbb{C}}$ matrice di perturbazione e consideriamo $\mu = \lambda(A + E)$, autovalore di $A + E$. Allora

$$\min_{i=1}^{n} |\mu - \lambda_i(A)| \leq K(S) * |E|$$

con la norma matriciale assoluta.

Una norma matriciale assoluta è tale che la norma di una matrice diagonale è

$$\max_{i=1}^{n} |A_{ii}|$$

Le norme 1,2, infinito sono norme assolute.

Caso particolare: supponiamo che A sia hermitiana, diagonalizzata tramite matrici unitarie Q, Q_h. Allora

$$|Q| = |Q_h| = 1$$

Allora $K(Q) = K(S) = 1$.

Se A è hermitiana, il problema del calcolo di autovalori è ben condizionato.

Se A non è hermitiana, allora il problema può essere ben condizionato o mal condizionato, e il condizionamento dipende dalla matrice S.

3.3 Metodo della potenza

3.3.1 Autovalore di modulo massimo

Supponiamo che l'autovalore di modulo massimo $|\lambda_1|$ sia strettamente maggiore dell'autovalore successivo, cioè

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$$

si esclude quindi il caso in cui λ_1 è un autovalore complesso (e avrebbe modulo uguale a quello del suo coniugato) o in cui λ_1 ha molteplicità maggiore di 1 (non sarebbe strettamente maggiore di se stesso).

Supponiamo che esistano $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ autovettori linearmente indipendenti, che formano una base.

Consideriamo un vettore

$$\boldsymbol{z}_0 = \sum_{j=1}^{n} \alpha_j \boldsymbol{x}_j, \quad \text{rappresentazione } \circ$$

rappresentato nella base degli autovettori. Richiedo che nella scrittura di $\boldsymbol{z}_0$ ci sia un contributo dell'autovettore $\boldsymbol{x}_1$ relativo all'autovalore λ_1, cioè $\alpha_1 \neq 0$.

Genero una successione di vettori $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k$ tali che

$$\boldsymbol{z}_k = A * \boldsymbol{z}_{k-1} = A^k * \boldsymbol{z}_0 \quad \text{da cui metodo della potenza}$$

Allora sostituendo $\boldsymbol{z}_0$ la rappresentazione $\circ$ ottengo:

$$\boldsymbol{z}_k = A^k [\sum_{j=1}^{n} \alpha_j \boldsymbol{x}_j]$$

e siccome gli $\boldsymbol{x}_j$ sono autovettori possiamo scrivere:

$$\boldsymbol{z}_k = \sum_{j=1}^{n} \lambda_j^k \alpha_j \boldsymbol{x}_j$$

$$\boldsymbol{z}_k = \lambda_1^k \alpha_1 \boldsymbol{x}_1 + \sum_{j=2}^{n} \lambda_j^k \alpha_j \boldsymbol{x}_j$$

e raccogliendo λ_1^k:

$$\boldsymbol{z}_k = \lambda_1^k [\alpha_1 \boldsymbol{x}_1 + \sum_{j=2}^{n} (\frac{\lambda_j}{\lambda_1})^k \alpha_j \boldsymbol{x}_j], \quad \text{rappresentazione } \heartsuit$$

$\lambda_j < \lambda_1 \longrightarrow (\frac{\lambda_j}{\lambda_1})^k \to 0 \iff k \to \infty$, allora per $k \to \infty$, $\boldsymbol{z}_k \parallel \boldsymbol{x}_1$.
Definisco il quoziente di Rayleigh:

$$\sigma_k = \frac{\boldsymbol{z}_k^h * A * \boldsymbol{z}_k}{\boldsymbol{z}_k^h \boldsymbol{z}_k}$$

Verifichiamo che $\sigma_k \to \lambda_1$. Sostituisco l'espressione $\heartsuit$ di $\boldsymbol{z}_k$ nel σ_k che abbiamo definito:

$$\sigma_k = \frac{[\bar{\lambda}_1^k (\bar{\alpha}_1 \boldsymbol{x}_1^h + \sum_{j=2}^{n} (\frac{\bar{\lambda}_j}{\lambda_1})^k \bar{\alpha}_j \boldsymbol{x}_j^h] * A * [\lambda_1^k (\boldsymbol{x}_1 \alpha_1 + \sum_{j=2}^{n} (\frac{\lambda_j}{\lambda_1})^k \alpha_j \boldsymbol{x}_j]}{[\bar{\lambda}_1^k (\bar{\alpha}_1 \boldsymbol{x}_1^h + \sum_{j=2}^{n} (\frac{\bar{\lambda}_j}{\lambda_1})^k \bar{\alpha}_j \boldsymbol{x}_j^h] * [\lambda_1^k (\alpha_1 \boldsymbol{x}_1 + \sum_{j=2}^{n} (\frac{\lambda_j}{\lambda_1})^k \alpha_j \boldsymbol{x}_j]}$$

Per $k \to \infty$ i termini $(\frac{\lambda_j}{\lambda_1})^k \to 0$, mentre i termini $\bar{\lambda}_1^k * \lambda_1^k$ si semplificano e rimane:

$$\lim_{k \to \infty} \sigma_k =$$

$$= \frac{\bar{\alpha}_1 \alpha_1 \boldsymbol{x}_1^h A \boldsymbol{x}_1}{\bar{\alpha}_1 \alpha_1 \boldsymbol{x}_1^h \boldsymbol{x}_1}$$

$$= \frac{\boldsymbol{x}_1^h \lambda_1 \boldsymbol{x}_1}{\boldsymbol{x}_1^h \boldsymbol{x}_1} = \lambda_1$$

Sul calcolatore il problema compare nella divisione per $\boldsymbol{x}_k^h \boldsymbol{x}_k = |\boldsymbol{x}_k|$, perché si rischia di andare nella barriera di *overflow* o di *underflow*. Per risolvere il problema, invece di considerare $\boldsymbol{z}_k$, lo si normalizza in norma 2 cioè si considera:

$$\boldsymbol{y}_k = \frac{\boldsymbol{z}_k}{|\boldsymbol{z}_k|}$$

e in questo modo nel quoziente di Rayleigh si divide per un vettore di norma 1.

3.3.2 Implementazione dell'algoritmo

Dato $\boldsymbol{z}_0$, eseguo le seguenti operazioni:

$$\boldsymbol{y}_k = \frac{\boldsymbol{z}_k}{|\boldsymbol{z}_k|}$$

$$\boldsymbol{z}_{k+1} = A \boldsymbol{y}_k$$

$$\sigma = \boldsymbol{y}_k^h \boldsymbol{z}_{k+1}$$

e se una certa condizione imposta nel test d'arresto non è verificata, si procede.

Il costo è dato da n^2 operazioni del prodotto matrice-vettore, e $3n$ operazioni per i prodotti scalari nelle norme.

$\sigma_k = \lambda_1 * (1 + O(\frac{\lambda_2}{\lambda_1})^k))$, cioè la velocità con cui σ_k converge a λ_1 è come il rapporto tra $\frac{\lambda_2}{\lambda_1}$, e se i due autovalori sono ben separati il metodo converge rapidamente.

Il metodo converge anche se l'autovalore di modulo massimo non ha molteplicità algebrica uguale a 1, in particolare:

1. $\lambda_1 = \lambda_2$. I due (o r) autovalori dominanti sono coincidenti. In questo caso il metodo risulta ancora convergente, ma x_k converge a un vettore nel sottospazio generato da x_1e x_2;
2. $\lambda_1 = -\lambda_2$. I due autovalori dominanti sono opposti. In questo caso l'autovalore di modulo massimo può essere approssimato applicando il metodo delle potenze alla matrice A^2, in modo che $\lambda_1^2 = \lambda_2^2$ e l'analisi ricada nel punto precedente;
3. i due autovalori dominanti sono complessi coniugati. In questo caso si hanno delle oscillazioni non smorzate nella sequenza dei vettori x_k e il metodo delle potenze non converge.

$\alpha_1 \neq 0$ non è un'ipotesi essenziale se si lavora in aritmetica *floating point*.

3.3.3 Test di arresto

Test dell'incremento: richiediamo che

$$|\sigma_k - \sigma_{k-1}| < \varepsilon$$

e il metodo prosegue fino a quando questa condizione non è verificata.

Nel caso in cui A è una matrice normale, cioè $AA_h = A_hA$, possiamo considerare altri due test d'arresto.

Proposizione 3.1

Supponiamo di avere $A \in \mathbb{C}_{n,n}$ normale, consideriamo $\boldsymbol{x} \in \mathbb{C}^n \setminus 0$, e f razionale definita su un sottoinsieme del piano complesso che contiene tutti gli autovalori di A. Allora esiste $\lambda \in A$ tale che

$$|f(\lambda)| \leq \frac{|f(A)\boldsymbol{x}|_2}{|\boldsymbol{x}|_2}$$

Test 1: Impongo la condizione

$$|\boldsymbol{z}_{k+1} - \sigma_k \boldsymbol{y}_k| \leq \varepsilon$$

che permette di avere un controllo sull'errore assoluto.

Infatti, se scegliamo la funzione $f(\lambda) = \lambda - \sigma_k$ (errore assoluto) e $\boldsymbol{x} = \boldsymbol{y}_k$, allora, per la proposizione:

$$|f(\lambda)| = |\lambda - \sigma_k| \leq \frac{|(A - \sigma_k I)\boldsymbol{y}_k|}{|\boldsymbol{y}_k|_2}$$

$|\boldsymbol{y}_k| = 1$ per costruzione, allora

$$|\lambda - \sigma_k| \leq |A * \boldsymbol{y}_k - \sigma_k I \boldsymbol{y}_k| = |\boldsymbol{z}_{k+1} - \sigma_k \boldsymbol{y}_k|$$

Test 2: se A è normale e non singolare, allora la condizione

$$\frac{|\boldsymbol{z}_{k+1} - \sigma_k \boldsymbol{y}_k|_2}{|\boldsymbol{z}_{k+1}|} \leq \varepsilon$$

permette di controllare l'errore relativo.

Scegliamo $f(\lambda) = \frac{\lambda - \sigma_k}{\lambda}$ (errore relativo), e poniamo $\boldsymbol{x} = \boldsymbol{z}_{k+1}$, applichiamo poi la proposizione:

$$\frac{|\lambda - \sigma_k|}{\lambda} \leq \frac{|(A - \sigma_k I)A^{-1}\boldsymbol{z}_{k+1}|_2}{|\boldsymbol{z}_{k+1}|}$$

$$= \frac{|\boldsymbol{z}_{k+1} - \sigma_k A^{-1}\boldsymbol{z}_{k+1}|}{|\boldsymbol{z}k + 1|_2}$$

$$= \frac{|\boldsymbol{z}_{k+1} - \sigma_k \boldsymbol{y}_k|}{|\boldsymbol{z}_{k+1}|_2}$$

3.4 Metodo della potenza inversa

3.4.1 Autovalore di modulo minimo

Il metodo della potenza inversa serve a calcolare l'autovalore e autovettore di modulo minimo. Si richiede che l'autovalore di modulo minimo sia strettamente maggiore di tutti gli altri, cioè

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_{n-1}| > |\lambda_n|$$

Considerando λ autovalore di A segue che:

$$A\boldsymbol{x} = \lambda\boldsymbol{x} \longrightarrow A^{-1}\lambda\boldsymbol{x} = \boldsymbol{x} \longrightarrow A^{-1}\boldsymbol{x} = \frac{1}{\lambda}\boldsymbol{x},$$

cioè se λ è autovalore di A, $1/\lambda$ è autovalore di A^{-1}.

Allora

$$\min_{i=1}^{n}\{|\lambda(A)|\} = \min_{i=1}^{n} \frac{1}{|\lambda_i(A^{-1})|}$$

$$= \frac{1}{\max\{\lambda_i(A^{-1})\}}$$

Ma per calcolare l'autovalore di modulo massimo si può applicare il metodo della potenza alla matrice inversa A^{-1}, e basta poi fare il reciproco dell'autovalore trovato.

3.4.2 Algoritmo del metodo

Dato $\boldsymbol{z}_0$, applichiamo il seguente algoritmo:

$$\boldsymbol{y}_k = \frac{\boldsymbol{z}_k}{|\boldsymbol{z}_k|}$$

$$\boldsymbol{z}_{k+1} = A^{-1}\boldsymbol{y}_k$$

$$\sigma_k = \boldsymbol{y}_k^h \boldsymbol{z}_{k+1}$$

$$test\, d'arresto$$

$$\lambda_{min} = \frac{1}{\sigma}$$

Si presenta il problema del calcolo dell'inversa A^{-1}, che è mal condizionato. Allora invece di calcolare il prodotto matrice per vettore

$$\boldsymbol{z}_{k+1} = A^{-1}\boldsymbol{y}_k$$

risolvo il sistema

$$A\boldsymbol{z}_{k+1} = \boldsymbol{y}_k$$

e per farlo uso la fattorizzazione $PA = LU$. I vettori $\boldsymbol{y}_k$ vengono trovato nelle successive iterazioni, ma la fattorizzazione $PA = LU$ dopo essere stata trovata nel primo passo viene sfruttata per tutte le iterazioni successive.

3.4.3 Calcolo dell'autovalore più vicino ad α

Supponiamo di avere un valore assegnato α: per calcolare l'autovalore di A più vicino a esso applichiamo il metodo delle potenze a B data da $(A - \alpha I)^{-1}$.

$$\lambda_i(B) = \frac{1}{\lambda_i(A - \alpha I)} = \frac{1}{\lambda_i(A) - \alpha}$$

otteniamo $\sigma_k \to \frac{1}{\lambda(A)-\alpha}$, allora $\lambda \to \frac{1}{\sigma_k} + \alpha$.

Algoritmo:

$$\boldsymbol{y}_k = \frac{\boldsymbol{z}_k}{|\boldsymbol{z}_k|}$$

$$\boldsymbol{z}_{k+1} = B\boldsymbol{y}_k \quad \text{passaggio problematico, sostituito con } (A - \alpha I)\boldsymbol{z}_{k+1} = \boldsymbol{y}_k$$

$$\sigma_k = \boldsymbol{y}_k^h \boldsymbol{z}_{k+1}$$

Test d'arresto:

$$\sigma_k = 1/\sigma_k + \alpha$$

3.5 Casi particolari

3.5.1 Caso 1: matrici hermitiane

Una matrice hermitiana è diagonalizzata attraverso matrici unitarie, cioè $A = Q\lambda Q_h$ con λ diagonale e $QQ_h = Q_hQ = I$, cioè abbiamo una base ortonormale di autovettori. Allora in questo caso:

$$\sigma_k = \boldsymbol{y}_k^h \boldsymbol{z}_{k+1} = \boldsymbol{y}_k^h A\boldsymbol{y}_k = \frac{\boldsymbol{y}_k^h A\boldsymbol{y}_k}{\boldsymbol{y}_h^k \boldsymbol{y}_k}$$

$$= \frac{(A^k \boldsymbol{y}_0)^h * (A^{k+1}\boldsymbol{y}_0)}{(A^k \boldsymbol{y}_0)^h (A^k \boldsymbol{y}_0)} \text{ formula } \heartsuit$$

e poniamo

$$\boldsymbol{y}_0 = \sum_{j=1}^{n} \alpha_j \boldsymbol{y}_j$$

e sostituendo questa scrittura nella formula $\heartsuit$ otteniamo:

$$= \frac{(A^k \sum_{j=1}^{k} \alpha_j \boldsymbol{y}_j)^h * (A^{k+1} \sum_{j=1}^{n} \alpha_j \boldsymbol{y}_j)}{(A^k \sum_{j=1}^{n} \alpha_j \boldsymbol{y}_j)^h (A^k \sum_{j=1}^{n} \alpha_j \boldsymbol{y}_j)}$$

e tenendo conto che $A^k \boldsymbol{y}_j = \lambda_j^k \boldsymbol{y}_j$ mettendo in evidenza λ_1^k:

$$\sigma_k = \frac{\lambda_1 |\lambda_1^k|^2 [\bar{\alpha}_1 \boldsymbol{y}_1^h + \sum_{j=2}^{n} \bar{\alpha}_j \frac{\bar{\lambda}_j}{\lambda_1}^k \boldsymbol{y}_j^h] * [\alpha_1 \boldsymbol{y}_1 + \sum_j \alpha_j (\frac{\lambda_j}{\lambda_1})^{k+1} \boldsymbol{y}_j]}{|\lambda_1^k|^2 [\bar{\alpha}_1 \boldsymbol{y}_1^h + \sum_j \bar{\alpha}_j \frac{\bar{\lambda}_j}{\lambda_1}^k \boldsymbol{y}_j^h] * [\alpha_1 \boldsymbol{y}_1 + \sum_{j=2}^{n} \alpha_j (\frac{\lambda_j}{\lambda_1})^k \boldsymbol{y}_j]}$$

Si può semplificare $(|\lambda_1|^k)^2$, inoltre, siccome gli autovettori costituiscono una base ortogonale, i prodotti misti del tipo $\boldsymbol{y}_i \boldsymbol{y}_j, i \neq j$ si annullano, e rimane solo:

$$\sigma_k = \frac{\lambda_1 [|\alpha_1|^2 \boldsymbol{y}_1^h \boldsymbol{y}_1 + \sum_{j=2}^{n} |\alpha_j|^2 (\frac{\lambda_j}{\lambda_1})^{2k} \frac{\lambda_j}{\lambda_1} \boldsymbol{y}_j^h \boldsymbol{y}_j]}{\lambda_1 * (\alpha_1^2 \boldsymbol{y}_1^h \boldsymbol{y}_1 + \sum_{j=2}^{n} |\alpha_j|^2 (\frac{\lambda_j}{\lambda_1})^{2k} \boldsymbol{y}_j^h \boldsymbol{y}_j)}$$

per l'ortonormalità $\boldsymbol{y}_j^h \boldsymbol{y}_j = 1$, allora

$$\sigma_k = \lambda_1 \frac{\alpha_1^2 (1 + \sum_{j=2}^{n} \frac{|\alpha_j|^2}{|\alpha_1|^2} * |\lambda_j/\lambda_1|^{2k} * |\lambda_j/\lambda_1|)}{|\alpha_1^2| * (1 + \sum_{j=2}^{n} |\frac{\alpha_j^2}{\alpha_1^2}| (\lambda_j/\lambda_1)^{2k})}$$

Al primo ordine, nella sommatoria sopravvive il termine per $j = 2$ (secondo autovalore più grande), quindi

$$\sigma = \frac{\lambda_1 [1 + |\alpha_2/\alpha_1|^2 |\lambda_2/\lambda_1|^{2k} |\lambda_2/\lambda_1|]}{[1 + |\alpha_2/\alpha_1|^2 |\lambda_2/\lambda_1|^{2k}]}$$

$$= \frac{\lambda_1 (1 + \varepsilon)}{1 + \bar{\varepsilon}}$$

$$= \lambda_1 * (1 + \varepsilon)(1 - \bar{\varepsilon})$$

$$\sigma_k = \lambda_1 [1 + |\alpha_2/\alpha_1|^2 (\lambda_2/\lambda_1)^{2k} (\lambda_2/\lambda_1)] * [(1 - |\alpha_2/\alpha_1|^2 |\lambda_2/\lambda_1|^{2k}]$$

$$= \lambda_1 * (1 - |\alpha_1/\alpha_2|^2 |\lambda_2/\lambda_1|^{2k})$$

Allora σ_k nel caso di matrici hermitiane:

$$\sigma_k = \lambda_1 (1 + O(|\lambda_2/\lambda_1|^{2k}))$$

A differenza del caso generale compare l'esponente $2k$ e la convergenza è più veloce.

Il metodo della potenza non converge su una matrice hermitiana quando la matrice ha autovalori $\alpha, -\alpha$ di moduli massimo. Escludiamo anche il caso di autovalori complessi

3.5.2 Caso 2: matrici hermitiane definite positive

Consideriamo $A = Q\lambda Q_h$ con Q unitaria e $\lambda \in \mathbb{R}^+$. Il caso $\alpha, -\alpha$ non può verificarsi, quindi su una matrice hermitiana definita positiva il metodo delle potenze converge sempre.

Risparmio del costo computazionale: cerchiamo $\lambda_{min}(A)$ con il metodo delle potenze inverse. Ad ogni iterazione si risolve il sistema lineare

$$A\boldsymbol{z}_{k+1} = \boldsymbol{y}_k$$

con costo di fattorizzazione $o(n^3)$. La fattorizzazione viene fatta una sola volta. Ad ogni iterazione, risolvo i sistemi triangolari. Se R è triangolare superiore piena il costo è $o(n^2)$. Se invece R è a banda, il costo è $o(n)$. Lo stesso vale se R è sparsa.

Siccome le iterazioni fatte sono numerose, si cerca di ridurre il costo di ogni iterazione a scapito del costo iniziale.

Trasformo la matrice A in una matrice tridiagonale B. A questo punto il costo della fattorizzazione di Cholesky è $o(n)$, perché R è bidiagonale, e anche il costo delle singole operazioni si riduce.

Passo da A generica a B tridiagonale tramite trasformazioni di Hausolder:

$$A_{k+1} = T_k^{-1} A_k T_k, \quad k = 1, \dots, n-2$$

La matrice di sinistra azzera la sottocolonna e quella di destra azzera la sottoriga di A in modo da ottenere una matrice tridiagonale. Questa operazione costa $o(n^3)$.

3.5.3 Condizionamento

Consideriamo il caso in cui la fattorizzazione di Cholesky non può avvenire (cioè λ_{min} molto piccolo e A definita positiva). Invece che considerare A, consideriamo $\hat{A} = A + cI$, che è ancora hermitiana e definita positiva. Si usa la tecnica di *shift*. Gli autovalori di $\hat{A}$ sono gli autovalori di A shiftati di c. $K_2(A) = \frac{\lambda_{max}}{\lambda_{mi}}$ ed è grande, invece si ha

$$K_2(A + c) = \frac{\lambda_{max}(A + c)}{\lambda_{min}(A + c)} = o(1)$$

$\sigma_k(\hat{A})$ converge all'autovalore minimo di $\hat{A}$, invece

$$\sigma_k(A) = \sigma_k(\hat{A}) - c$$

e si ha come rischio l'errore di cancellazione. Bisogna bilanciare c rispetto a $\lambda_{min}(A)$.

Esercizio 3.1

Nel caso di matrici hermitiane, calcolare la velocità di convergenza nel caso di autovalore multiplo.

Sia λ_{max} di molteplicità r.

$$|\lambda_1| = |\lambda_2| = \lambda_r| > |\lambda_{r+1}| \geq \cdots \geq |\lambda_n|$$

$$\boldsymbol{y}_0 = \sum_{i=1}^{r} \alpha_{1,i} \boldsymbol{x}_{1,i} + \sum_{j=r+1}^{n} \alpha_j \boldsymbol{x}_j$$

$$\sigma_k = \lambda_1(1 + O(|\frac{\lambda_{r+1}}{\lambda_1}|^{2k})$$

si ritrova lo stesso risultato di prima con λ_{r+1} che sostituisce λ_2, cioè, al posto di λ_2 compare il primo valore diverso da λ_1.

4

Radici di equazioni non lineari

4.1 Osservazioni introduttive

4.1.1 Problema da risolvere

Data una funzione $f\colon [a,b] \to \mathbb{R}$, si cerca α tale che $f(\alpha) = 0$. Si utilizzano metodi iterativi, cioè si cercano x_k tali che

$$\lim_{k\to\infty} x_k = \alpha$$

La convergenza del metodo non dipende solo dal metodo usato, ma anche dalla scelta del vettore d'innesco: infatti se il vettore d'innesco appartiene a un determinato intorno della radice, il metodo converge, altrimenti questa condizione non è verificata.

I metodi usati possono avere una convergenza lineare, del tipo $E_k \leq c * E_{k-1}$ in cui sono necessarie varie iterazioni prima che l'ordine di grandezza dell'errore diminuisca. Altrimenti può verificarsi una convergenza superlineare: consideriamo un esempio di convergenza quadratica, in cui a un certo passo $E_k = 10^{-2}$, al successivo $E_{k+1} = 10^{-4}$ e al successivo $E_{k+2} = 10^{-8}$, cioè l'errore crolla velocemente.

4.1.2 Ordine di convergenza

$\Diamond$ **Definizione** Consideriamo una successione $x_k \to \alpha$, con $x_k \neq \alpha \forall k$. Diciamo che la successione ha **ordine di convergenza** P se esiste un $P \geq 1$ tale che

$$\lim_{k\to\infty} \frac{|x_{k+1} - \alpha|}{(|x_k - \alpha|)^P} = \gamma$$

dove $0 < \gamma < 1$ se $P = 1$, mentre $\gamma > 1$ se $P > 1$. γ si dice **fattore di convergenza**.

1. Se $P = 1$ e $\gamma < 1$ abbiamo una **convergenza lineare**;
2. se $P = 1$ e $\gamma = 1$ abbiamo una **convergenza sublineare**;
3. se $P > 1$ e $\gamma > 0$ abbiamo una **convergenza superlineare**.

In ogni caso, esiste una costante $c > 0$ tale che

$$|x_{k+1} - \alpha| \leq c|x_k - \alpha|^P$$

Per k abbastanza grande, cioè, in termini dell'errore assoluto:

$$E_{k+1} \leq cE_k^P$$

e in termini di errore relativo:

$$E_{r,k+1} = \frac{|x_{k+1} - \alpha|}{\alpha} \leq c\frac{|x_k - \alpha|^P}{\alpha}$$

$$\leq c|\alpha^{P-1}|(\frac{|x_k - \alpha|}{\alpha})^P \leq c\alpha^{P-1}E_{r,k}^P$$

Se $P = 1$, più è piccola la costante γ più il metodo converge rapidamente.

Supponiamo di avere $P = 2$. Allora se $E_k = 10^{-2}$, al passo successivo si ha $E_{k+1} = (10^{-2})^2 = 10^{-4}$, e $E_{k+2} = (10^{-4})^2 = 10^{-8}$. Se invece $P = 1$, $E_{k+1} = c * 10^{-2}$, dove $0 < c < 1$ e ci vogliono più iterazioni prima che cambi l'ordine dell'errore.

4.1.3 Esempi sui tipi di convergenza

Esempio 4.1 (convergenza lineare)

Consideriamo la successione

$$x_{k+1} = \cos x_k$$

con $P = 1$. Allora $\gamma = \sin \alpha$.

Esempio 4.2 (convergenza sublineare)

In questo tipo di convergenza la velocità di convergenza diminuisce quando ci si avvicina ad α. Consideriamo la successione:

$$x_{k+1} = \sin x_k$$

$$\gamma = 1$$

Esempio 4.3 (convergenza superlineare)

Consideriamo la successione:

$$5x_k^3 + x_k^4,\ P = 3,\ x \in (0, 1/3)$$

allora

$$\gamma = \lim_{k\to\infty} \frac{|x_{k+1} - \alpha|}{|x_k - \alpha|^P} =$$

$$\gamma = \lim_{k\to\infty} \frac{|5x_k^3 + x_k^4 - \alpha|}{|x_k - \alpha|^3} =$$

$$\gamma = 5$$

4.2 Bisezione corde secanti Newton

Data una funzione $f\colon [a, b] \to \mathbb{R}$, trovare α tale che $f(\alpha) = 0$.

4.2.1 Metodo di bisezione

Teorema 4.1 (teorema degli zeri per funzioni continue)

Data una funzione $f\colon [a,b] \to \mathbb{R}$ con $f \in C^0(a,b)$ se $f(a) * f(b) < 0$, allora esiste $\alpha \in (a,b)$ tale che $f(\alpha) = 0$.

◊ **Definizione** Diciamo che a, b costituiscono una bracket, cioè un intervallo contenente la radice, se $f(a) * f(b) < 0$.

Idea del metodo di bisezione: abbiamo una *bracket* di partenza (a,b), tale che $f(a) * f(b) < 0$, e cerco una *bracket* più piccola, e così via.

I passi del metodo sono i seguenti: calcolo $c = (a+b)/2$ punto medio di (a,b) (sul calcolatore per questioni di stabilità il punto medio si calcola come $c = a + (b-a)/2)$). Trovato c, si possono verificare tre casi:

1. se $f(a) * f(c) < 0$ allora la nuova bracket è l'intervallo (a,c);
2. se invece $f(c) * f(b) < 0$, allora la bracket è (b,c);
3. nel caso $f(c) = 0$, c è la radice cercata.

A partire da una bracket T_0, genero una sottosuccessione di intervalli I_k di estremi (a_k, b_k) con $k \geq 0$ con la proprietà che $I_k \subset I_{k-1}$, e tali che per ogni $f(a_k) * f(b_k) < 0$. Viene contemporaneamente generata una successione di valori C_k punti medi degli intervallini I_k, che converge alla radice.

A ogni iterazione, la distanza tra il punto medio C_k e α è minore di metà dell'ampiezza della *bracket*. Sia $L_0 = |b-a|$ l'ampiezza della *bracket* iniziale. Allora

$$L_k = d(a_k, b_k) = \frac{L_0}{2^k}.$$

Di conseguenza possiamo affermare

$$|C_k - \alpha| \leq 1/2 L_k = \frac{L_0}{2^{k+1}}$$

Per ottenere un'approssimazione sufficientemente buona della radice si richiede che $\frac{L_0}{2^{k+1}} \leq \varepsilon$.

In termini di iterazioni segue che:

$$2^{k+1} \geq \frac{L_0}{\varepsilon} \longrightarrow \bar{k} \geq \log_2(\frac{L_0}{\varepsilon}) - 1$$

dove $\bar{k}$ è il numero di iterazioni necessarie per ottenere un'approssimazione della radice a meno di ε. Il numero di iterazioni richieste dal metodo dipende solo dall'ampiezza della *bracket* iniziale.

La convergenza del metodo di bisezione è lenta, "lineare", e il metodo converge sempre. Questo metodo viene usato in mancanza di informazioni preliminari su dove si trova la radice, però non è applicabile a sistemi di equazioni non lineari. Per guadagnare una cifra decimale occorrono $3.3 = \log_2 10$ iterazioni.

4.2.2 Struttura generale degli altri metodi

Consideriamo la funzione f e lo sviluppo di Taylor centrato in α.

$$f(x) = f(\alpha) + (x - \alpha) * f'(\xi),\ \xi \in (\alpha, x)$$

che si riscrive come:

$$0 = f(\alpha) = f(x) - (\alpha - x) * f'(\xi)$$

Definiamo il metodo iterativo nel seguente modo: assegnato x_0, per ogni $k \geq 0$ determiniamo la nuova iterata x_{k+1} risolvendo l'equazione $\circ$:

$$f(x_k) + (x_{k+1} - x_k) * m_k = 0$$

dove m_k è un'opportuna approssimazione di $f'(x_k)$.

Trovare la soluzione di $\circ$ equivale a trovare l'intersezione tra l'asse x e la retta $y = m_k * (x - x_k) + f(x_k)$.

Esplicitando x_{k+1}, la relazione $\circ$ diventa

$$x_{k+1} = x_k - m_k^{-1} f(x_k), \quad k \geq 0$$

A seconda dell'approssimazione di $f'(x_k)$ e quindi della scelta di m_k si ottengono i metodi delle corde, secanti e di Newton.

4.2.3 Metodo delle corde

Scelta del coefficiente

Per ogni $k \geq 0$, scegliamo il coefficiente $m_k = m$ ponendo

$$m = \frac{f(b) - f(a)}{b - a},$$

quindi la relazione $\circ$ diventa:

$$x_{k+1} = x_k - \frac{b - a}{f(b) - f(a)} * f(x_k), \quad k \geq 0$$

Caratteristiche

A differenza del metodo di bisezione, qui a ogni iterata si usa il vettore ottenuto all'iterata precedente.

Ordine di convergenza

Questo metodo ha un ordine di convergenza di $P = 1$, quindi la convergenza è lineare.

Interpretazione gemetrica

Supponiamo di avere una funzione su un intervallo $[a, b]$, consideriamo un punto $x_0 \in (\sqrt{\alpha}, b)$, consideriamo la retta per $(x_0, f(x_0))$ con coefficiente angolare m definito come sopra.

L'iterata x_1 è l'intersezione di questa retta con l'asse delle ascisse. Consideriamo poi la retta per $(x_1, f(x_1))$ parallela a quelle disegnate sopra, e l'intersezione con l'asse x è l'iterata x_2.

Si considerano quindi rette tra loro parallele e ognuna passante per il punto $(x_k, f(x_k))$.

4.2.4 Metodo delle secanti

Scelta del coefficiente

Per ogni $k \geq 0$, fisso il coefficiente m_k ponendo

$$m_k = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

quindi la relazione $\circ$ diventa:

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} * f(x_k), \quad \forall k \geq 0$$

Caratteristiche

Questo non è un metodo del punto fisso, perché a ogni passo si considerano informazioni ottenute nelle due iterate precedenti. Tranne nel primo passo, in cui f viene valutata nei due valori di innesco, a ogni iterazione viene fatta una sola valutazione della funzione, e quindi, rispetto al metodo delle corde, il costo computazionale non aumenta di molto, aumenta solo nella prima iterazione.

Ordine di convergenza

Questo metodo è superlineare, perché $P > 1$.

In particolare vale il seguente teorema.

Teorema 4.2

Se f è C^2 in un opportuno intorno della radice, se $f'(\alpha) \neq 0$ cioè se α è una radice semplice, e se $x_0, x_1 \in U(\alpha)$, allora il metodo delle secanti converge con ordine $P = (1 + \sqrt{5})/2 = 1.6$, cioè con una convergenza superlineare.

Interpretazione geometrica

A differenza del grafico precedente qui le rette non sono parallele. Dopo aver tracciato la retta per $(x_k, f(x_k))$ e $(x_{k-1}, f(x_{k-1})$, l'iterata successiva x_{k+1} è l'intersezione di questa retta con l'asse x.

4.2.5 Metodo di Newton

Scelta del coefficiente

Si richiede che f sia di classe C^1 in un opportuno intorno della radice, che $f'(\alpha) \neq 0$, e per ogni k, fissiamo il coefficiente $m_k = f'(x_k)$. Allora

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k \geq 0$$

Caratteristiche

Il costo computazionale viene aumentato in modo considerevole perché viene richiesta anche la valutazione della funzione e della sua derivata prima in x_k.

Ordine di convergenza

Sotto opportune ipotesi di regolarità, il metodo permette di avere ordine di convergenza 2.

Interpretazione geometrica

Graficamente viene considerata la retta tangente alla funzione nel punto $(x_k, f(x_k))$, e si considera la sua intersezione con l'asse x che è l'iterata successiva.

4.3 Metodo di Brent

4.3.1 Struttura del metodo

È una combinazione del metodo di bisezione e delle secanti, in cui si sfruttano la sicurezza di convergenza del metodo di bisezione e la velocità di convergenza di quello delle secanti.

Il metodo richiede come input una *bracket* (b, c) con $f(b) * f(c) < 0$. Restituisce come output lo zero della funzione nella variabile chiamata b. Nell'algoritmo, b rappresenta l'approssimazione della radice cercata. Il metodo si basa sull'idea che se

$$|f(b)| < |f(c)|$$

allora b è più vicino alla radice di quanto lo sia c. Questo in realtà non è sempre vero, infatti se consideriamo una funzione che interseca l'asse x in $(0, \alpha)$ e poi decresce lentamente dopo tale punto, si ha che, pur scegliendo b lontano dalla radice, segue che $|f(b)| < |f(c)|$ ma c è più vicina alla radice di quanto lo sia b.

Nell'algoritmo, a, b rappresentano le due iterate con cui il metodo delle secanti calcola l'iterata successiva, cioè $b = x_k$, $a = x_{k-1}$, mentre b e c sono gli estremi della *bracket*.

4.3.2 Algoritmo

1. **Definizione di** a, b, c: in base all'idea precedente, se $|f(b)| > |f(c)|$ allora scambio fra loro b e c. Poniamo poi $a = c$ per far partire il metodo delle secanti.

2. **Test d'arresto sull'ampiezza della *bracket***: finché $|b - c| > \varepsilon$ le istruzioni vengono eseguite.

3. **Nuovo passo**: ci si avvicina alla radice attraverso l'assegnamento

$$b = b + d_d$$

dove d_d può essere posto uguale al passo del metodo delle secanti oppure uguale al passo del metodo di bisezione, e questa scelta viene valutata a ogni iterazione.

4. **Valutazione dei due passi possibili**: per scegliere quali dei due metodi usare, si valuta di quanto ci si sposta da b in entrambi i metodi:

$$d_m = (c - b)/2 \quad \text{passo di bisezione}$$

Prima di calcolare d_s (passo del metodo delle secanti), si valuta $d_f = f(b) - f(a)$. Se $d_f = 0$, si pone $d_d = d_m$, perché d_s non è applicabile. Altrimenti

$$d_s = a - \frac{b-a}{f(b) - f(a)} * f(b)$$

5. **Scelta del passo**: se entrambi i passi sono applicabili, si eseguono le seguenti istruzioni:

$$\text{if}(\text{sgn}d_m == \text{sgn}d_s \wedge |d_s| <= |d_m|)$$

$$d_d = d_s$$

$$\text{else}$$

$$d_d = d_m$$

$$\text{end}$$

Nota: queste istruzioni si giustificano nel seguente modo. Il passo di bisezione rimane sempre nella *bracket* per definizione del metodo, invece, se il passo di secanti ha direzione opposta, si esce dalla *bracket*. Una volta assicurato che entrambi i metodi stanno dentro la *bracket*, si sceglie il passo di ampiezza più piccola, che fa allontanare di meno dall'approssimazione corrente della radice.

6. **Controllo**[1]

7. **Applicazione del passo scelto**: $d = b + d_d$

8. **Preparazione della terna di punti per un eventuale passo successivo**:

$$a_1 = a$$

$$b_1 = d$$

$$\text{if}(\text{sgn}f(d) \neq \text{sgn}f(c))$$

$$c_1 = c$$

$$\text{else}$$

$$c_1 = b$$

[1]Prima di usare il passo scelto, bisogna fare un controllo. Se si continua ad applicare il metodo delle secanti, l'ampiezza della *bracket* non si riduce, e il metodo continua a fare iterazioni anche vicino alla radice, perché la condizione del test di arresto sulla bracket non viene verificata. Allora si aggiunge il seguente test: se $|d_d| < \varepsilon$ (quindi se il metodo delle secanti sta convergendo), si pone:

$$d = \varepsilon/2 * \text{sgn}d_d$$

Nel momento in cui d scavalca la radice, la bracket si riduce velocemente e il metodo si blocca.

end

Nota: se $\text{sgn} f(d) \neq \text{sgn} f(c)$, allora segue necessariamente che $\text{sgn} f(d) \neq \text{sgn} f(b)$, perché b, c erano una bracket)

9. **Assegnamento**:

$$b = b_1$$

$$c = c_1$$

$$\text{if}(|\text{f(b)}|>|\text{f(c)}|)$$

$$temp = b;$$

$$b = c;$$

$$c = temp;$$

$$\text{end}$$

$$a_1 = c$$

Metodo implementato:

```
function [b,it] = Mybrent(f,a,b,c)
if(abs(feval(f,b))>=abs(feval(f,c)))
temp = b;
b = c;
c = temp;
\noindent end
while(abs(b-c)>eps)
\noindent dm = (c-b)/2;
q = feval(f,a)-feval(f,b);
if(q==0)
\noindent ds = dm;
\noindent else
\noindent ds = a-(b-a)/(feval(f,b)-feval(f,a))*feval(f,b);
\noindent end
if(sign(ds)==sign(dm)&&abs(ds)<=abs(dm))
\noindent dd = ds;
\noindent else
\noindent dd = dm;
\noindent end
if(dd<eps)
\noindent d = eps/2*sign(dd);
\noindent end
```

```
\noindent d = b+dd;
b1 = d;
if(sign(b1)==sign(b))
c1 = c;
\noindent else
c1 = b;
\noindent end
b = b1;
c = c1;
if(abs(feval(f,c))<=abs(feval(f,b)))
temp = b;
b = c;
c = temp;
\noindent end
\noindent end
```

4.4 Metodi del punto fisso

4.4.1 Riformulazione del problema

Data $f\colon (a,b) \to \mathbb{R}$ trovare α tale che $f(\alpha) = 0$ equivale a cercare α tale che $\alpha = \phi(\alpha)$ con ϕ funzione opportuna. Ad esempio, si può considerare

$$\phi = x + f(x)$$

oppure

$$\phi(x) = x + \frac{f(x)}{h(x)}, \quad h(\alpha) \neq 0$$

oppure

$$\phi(x) = x + F(f(x))$$

con F continua e $F(0) = 0$.

4.4.2 Esistenza di punti fissi

Teorema 4.3

Se ϕ mappa un intervallo $[a,b]$ in sé stesso, se $\phi \in C^1([a,b])$, e se esiste un $K < 1$ tale che $|\phi'(x)| \leq K \forall x \in [a,b]$, segue che:

1. ϕ ha un unico punto fisso nell'intervallo $[a,b]$;
2. $x_k = \phi(x_{k-1})$ tende ad α per qualunque punto d'innesco $x_0 \in [a,b]$;
3. $\lim_{k\to\infty} \frac{x_{k+1}-\alpha}{x_k-\alpha} = \phi'(\alpha)$.

Possiamo definire il tasso asintotico di convergenza come $-\log|f'(\alpha)|$. La convergenza è globale nell'intervallo $[a,b]$.

Dimostrazione

Esistenza del punto fisso: per ipotesi, $\phi\colon [a,b] \to [a,b]$ è continua. Data $\psi(x) = \phi(x) - x\colon [a,b] \to [a,b]$, siccome $a < \phi(x) < b$, si ha $\psi(a) = \phi(a) - a \geq 0$ e $\psi(b) = \phi(b) - b < 0$. Quindi ψ è una funzione continua con segni opposti agli estremi, allora esiste $\alpha \in [a,b]$ tale che $\psi(\alpha) = \phi(\alpha) - \alpha = 0$, cioè $\phi(\alpha) = \alpha$.

Unicità del punto fisso: per assurdo, supponiamo che esistano α_1, α_2 tali che $\phi(\alpha_1) = \alpha_1$ e $\phi(\alpha_2) = \alpha_2$, allora

$$|\alpha_2 - \alpha_1| = |\phi(\alpha_2) - \phi(\alpha_1)| = \phi'(\eta) * (\alpha_2 - \alpha_1), \quad \eta \in [\alpha_1, \alpha_2].$$

Per l'ipotesi $\phi'(x) < k$ segue che

$$|\alpha_2 - \alpha_1| \leq K * |\alpha_2 - \alpha_1| < |\alpha_2 - \alpha_1|$$

e abbiamo un assurdo.

Convergenza della successione delle iterate: per ogni $k \geq 0$, esiste un punto $\eta_k \in [\alpha, x_k]$ tale che

$$x_{k+1} - \alpha = \phi(x_k) - \phi(\alpha) = \phi'(\eta_k) * (x_k - \alpha), \quad \alpha < \eta_k < x_k \quad \text{formula } *$$

e passando al valore assoluto

$$|x_{k+1} - \alpha| \leq K * |x_k - \alpha| \leq K^{k+1} * |x_0 - \alpha|$$

e siccome $K \leq 1$, la quantità tende a 0 per $k \to \infty$, cioè $\{x_k\} \to \alpha$, per ogni $x_0 \in [a,b]$.

Verifica del limite: per la formula $*$:

$$\lim_{k\to\infty} \frac{x_{k+1} - \alpha}{x_k - \alpha} = \lim_{k\to\infty} \phi'(\eta_k) = \phi'(\alpha)$$

4.4.3 Interpretazione geometrica dei metodi di punto fisso

Consideriamo il metodo

$$x_{k+1} = \phi(x_k)$$

Nel caso di convergenza, ci sono due situazioni possibili:

1. convergenza monotona: i punti x_k si trovano sempre a sinistra o sempre a destra della radice;
2. convergenza alternata: se x_k si trova a sinistra della radice, x_{k+1} si trova a destra e così via.

Consideriamo il punto di intersezione di ϕ con la bisettrice, che è il punto fisso α. Consideriamo x_0 a sinistra di α, e poniamo $\phi(x_0) = x_1$. Per determinare $x_2 = \phi(x_1)$ tracciamo a partire da x_1 il segmento orizzontale che lo congiunge alla bisettrice, e poi risaliamo sulla funzione, trovando $x_2 = \phi(x_1)$ e così via. Se $x_0 < \alpha$ e la funzione è crescente abbiamo una successione monotona crescente, mentre è decrescente nell'altro caso.

4.4.4 Condizione sufficiente di convergenza locale

Teorema 4.4 (teorema di Ostrosky)

Sia α un punto fisso di ϕ, e supponiamo che ϕ sia di classe C^1 in un intorno del punto fisso. Allora esiste $\delta > 0$ tale che $\forall x_0 t.c. |x_0 - \alpha| < \delta$, la successione converge e

$$\lim_{k\to\infty} x_k = \alpha.$$

Si sa che il δ esiste, ma non si sa come costruirlo.

Proposizione 4.1

Sia ϕ di classe C^{p+1} in un opportuno intorno di α, e supponiamo che $\phi^s(\alpha) = 0$ con $s = 1, \ldots, p$, mentre $\phi^{s+1}(\alpha) \neq 0$, allora il metodo di punto fisso ha ordine di convergenza $p + 1$, e

$$\lim_{k\to\infty} \frac{x_{k+1} - \alpha}{(x_k - \alpha)^{p+1}} = \frac{\phi^{p+1}(\alpha)}{(p+1)!}$$

4.4.5 Osservazione sul metodo di Newton

Il metodo di Newton è un metodo di punto fisso, dove si pone:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

quindi ϕ è della forma $\phi(x) = x - \frac{f(x)}{h(x)}$.

Se $f'(\alpha) \neq 0$ e quindi se α è una radice semplice, allora per la proposizione sopra $\gamma = 1/2!\phi''(\alpha)$, e si ha

$$\phi'(x) = 1 - \frac{f'(x)^2 - f(x) * f''(x)}{f'(x)^2}$$

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

e vale 0 se valutato in α.

$$\phi''(x) = \frac{f'(x)^2 * (f(x) * f^{(3)}(x) + f'(x) * f''(x)) - 2 * f'(x) * f''(x) * f(x) * f''(x)}{f'(x)^4}$$

e valutando in α, tenendo conto che $f(\alpha) = 0$:

$$\phi''(x) = \frac{f'(\alpha)^3 * f''(\alpha)}{f'(\alpha)^4} = \frac{f''(\alpha)}{f'(\alpha)}$$

quindi

$$\lim_{k\to\infty} \frac{x_{k+1} - \alpha}{(x_k - \alpha)^2} = 1/2 \frac{f''(\alpha)}{f'(\alpha)}$$

Se α è una radice multipla di molteplicità s, si può scrivere $f(x) = (x - \alpha)^s * \nu(x)$. Ripetendo i conti precedenti,

$$\phi'(x) = \frac{f(x) * f''(x)}{f'(x)^2}$$

$$f'(x) = s(x-\alpha)^{s-1} * \nu(x) + (x-\alpha)^s * \nu'(x) = (x-\alpha)^{s-1} * [s * \nu(x) + (x-\alpha) * \nu'(x)]$$

$$f''(x) = (x-\alpha)^{s-1} * [s * \nu'(x) + +(-\alpha) * \nu'(x) + (x-\alpha) * \nu''(x)] + (s-1) * (x-\alpha)^{s-2} * [s * \nu(x) + (x-\alpha) * \nu'(x)]$$

$$f''(x) = (x-\alpha)^{s-2} * [(x-\alpha) * [s * \nu'(x) - \alpha * \nu'(x) + (x-\alpha) * \nu''(x) + (s-1) * [s * \nu(x) + (x-\alpha) * \nu'(x)]]$$

$$f''(x) = (x-\alpha)^{s-2} * [(x-\alpha) * [(s-\alpha) * \nu'(x) + (x-\alpha) * \nu''(x) + (s-1) * s * \nu(x) + (s-1)(x-\alpha) * \nu'(x)]]$$

$$\phi''(x) = \frac{(x-\alpha)^s * \nu(x) * (x-\alpha)^{s-2} * \theta}{((x-\alpha)^{s-1})^2 * \eta^2} = \frac{\theta}{\eta^2}$$

con

$$\eta = s * \nu(x) + (x-\alpha) * \nu'(x)$$

$$\eta(\alpha) = s * \nu(\alpha)$$

$$\theta = [(x-\alpha) * [(s-\alpha) * \nu'(x) + (x-\alpha) * \nu''(x) + (s-1) * s * \nu(x) + (s-1)(x-\alpha) * \nu'(x)]]$$

$$\theta(\alpha) = (s-1) * s * \nu(\alpha)$$

quindi

$$\phi''(\alpha) = \frac{\theta(\alpha)}{\eta^2(\alpha)} = \frac{(s-1)s * \nu(\alpha)}{s^2 * \nu^2(\alpha)}$$

$$= (s-1)/s * \nu(\alpha)$$

e quindi $f'(\alpha) = 0$ solo se $s = 1$, e quindi se α è una radice semplice.

Nel caso di radici multiple, per recuperare l'ordine di convergenza, costruisco la funzione $\phi(x) = x - s\frac{f(x)}{f'(x)}$, per ripristinare la convergenza superlineare, infatti si ottiene:

$$\phi'(\alpha) = 1 - s + s\frac{f(\alpha)f''(\alpha)}{f'(\alpha)^2}$$

e sostituendo il valore $\phi'(\alpha) = (s-1)/s$ trovato prima:

$$\phi''(\alpha) = 1 - s + s * \phi'(\alpha) = 1 - s + s(s-1)/s = 0$$

4.5 Condizionamento test e stabilità dei metodi di punto fisso

4.5.1 Condizionamento

Un problema è ben condizionato se, variando di poco i dati del problema, si ottengono piccole variazioni nel risultato. Per studiare il condizionamento del metodo, supponiamo quindi di avere $f(x) = \phi(x) - D$, con D piccola perturbazione della ϕ originaria.

Supponiamo che la funzione ϕ sia invertibile, allora $f(\alpha) = 0$ se e solo se $\phi(\alpha) = D$, e quindi $\alpha = \phi^{-1}(D)$. Sia $G_D = \phi^{-1}(D) = \alpha$, e α la radice di ϕ. L'espressione per l'indice di condizionamento è:

$$I_{cond} = \frac{D}{G_D} * (G_D)'$$
$$= \frac{D}{\alpha} * (\phi^{-1})' = \frac{D}{\alpha} * \frac{1}{\phi'(\alpha)}$$

allora il problema è ben condizionato se $\phi'(\alpha)$ è grande, mentre è mal condizionato se $\phi'(\alpha)$ è piccolo. Questo equivale a dire che nel caso di una funzione "piatta" nell'intorno della radice (cioè $\phi'(\alpha)$ piccolo) il problema è mal condizionato.

4.5.2 Test del residuo

Data la successione di iterate $\{x_k\}$ generate dal metodo, a ogni iterata si calcola $|\phi(x_k)|$. Se $|f(x_k)| < \varepsilon$ con ε fissato, l'approssimazione della radice è già buona e il metodo si arresta. Facciamo l'ipotesi che f sia di classe C^1, allora, siccome $f(\alpha) = 0$ si ha:

$$f(x_k) = f(x_k) - f(\alpha) = f'(\eta_k) * (x_k - \alpha), \quad \alpha < \eta_k < x_k,$$

con $\eta_k - \alpha < x_k - \alpha$, dove l'ultimo passaggio deriva dall'applicazione del teorema del valor medio.

Segue che $x_k - \alpha = \frac{f(x_k)}{f'(\eta_k)}$, e passando al valore assoluto e supponendo $|f(x_k)| \leq \varepsilon$ si ha:

$$|x_k - \alpha| \leq \frac{\varepsilon}{f'(\eta_k)}$$

e quindi l'errore assoluto è ben controllato dal residuo.

Supponiamo che la successione di iterate $\{x_k\}$ converga ad α, quindi $f'(\eta_k) \to f'(\alpha)$, e il modo con cui il residuo controlla l'errore assoluto dipende anche da $f'(\alpha)$. Considerando

$$f(\alpha) \leq \frac{\varepsilon}{f'(\alpha)},$$

segue che:

1. se $|f'(\alpha)| = 1$ il test è buono;
2. se $|f'(\alpha)| < 1$ il termine a destra cresce molto, e il test non dà informazioni, perché a un residuo piccolo può corrispondere un errore grande. Si conclude che il test del residuo non dà informazioni quando il problema è mal condizionato $f'(\alpha)$ piccolo);
3. se $|f'(\alpha)| >> 1$, il test è troppo restrittivo, perché pur avendo già trovato una buona approssimazione della radice, il metodo non si ferma perché il test del residuo non è soddisfatto.

4.5.3 Test dell'incremento

Calcolo le iterate finché

$$|x_{k+1} - x_k| \leq \varepsilon$$

Supponiamo di generare una successione di iterate tramite una generica funzione ϕ, e supponiamo che ϕ sia di classe C^1. Tenendo conto della definizione delle iterate si può scrivere:

$$x_{k+1} - \alpha = \phi(x_k) - \phi(\alpha) = \phi'(\eta_k) * (x_k - \alpha)$$

con $|\eta_k - \alpha| < |x_k - \alpha|$ per il teorema del valor medio. Determiniamo la relazione tra l'errore relativo e l'incremento:

$$x_{k+1} - x_k = x_{k+1} - \alpha - (x_k - \alpha) = \phi'(\eta_k) * (x_k - \alpha) - (x_k - \alpha) = (x_k - \alpha) * (\phi'(\eta_k) - 1)$$

quindi

$$x_k - \alpha = \frac{x_{k+1} - x_k}{\phi'(\eta_k) - 1}$$

e in valore assoluto l'errore assoluto è controllato da

$$\frac{\varepsilon}{|\phi'(\eta_k) - 1|}$$

A differenza del caso precedente:

1. se $\phi'(\alpha) \simeq 1$ il test non dà informazioni;
2. se $\phi'(\alpha) = 0$ si è nel caso favorevole, perché l'incremento controlla l'errore assoluto (bilanciamento ottimale).

Si considera un metodo superlineare con $p \geq 2$ (ad esempio il metodo di Newton nel caso in cui α sia semplice). Per questo metodo il test dell'incremento è buono.

4.5.4 Stabilità

Supponiamo di voler calcolare la successione di iterate $\{x_{k+1}\}$ a partire da un punto di innesco, con la legge $x_{k+1} = \phi(x_k)$, e indichiamo con $\{\bar{x}_k\}$ la successione delle iterate effettivamente ottenute sul calcolatore.

Può succedere che $\bar{x}_k$ non convergono ad α, pur appartenendo a un intorno della radice di raggio via via più piccolo fino a una certa soglia.

Proposizione 4.2

Nelle ipotesi precedenti, supponiamo

$$\bar{x}_k = \phi(\bar{x}_{k-1}) + \delta_k$$

con δ_k errore commesso localmente. Assumo per comodità che tutti i δ_k siano maggiorati da un certo δ. Sia α un punto fisso di ϕ, $\alpha \in (\alpha - \rho, \alpha + \rho)$, e sia $\lambda = \max_{|x-\alpha| \leq \rho} |\phi'(x)|$. Se definisco $\sigma = \frac{\delta}{1-\lambda} < \rho$, consideriamo un qualsiasi x_0 tale che $d(\alpha, x_0) < \rho$, e supponiamo per comodità che $\bar{x}_0 = x_0$ (assenza di errore iniziale), allora segue che

$$|\bar{x}_k - \alpha| \leq \sigma + \lambda^k(\rho - \sigma), \forall k \geq 0$$

Dim. Dimostro per induzione.
1. Se $k = 0$, la tesi è vera perché $|\bar{x}_0 - \alpha| \leq \rho$ per ipotesi, e $\rho = \sigma + \lambda^0(\rho - \sigma)$.
2. Ipotesi induttiva: $|\bar{x}_{k-1} - \alpha| \leq \sigma + \lambda^{k-1}(\rho - \sigma)$
3. Dimostriamo l'asserto per k:

$$|\bar{x}_k - \alpha| = |\phi(\bar{x}_{k-1}) + \delta_k - \alpha| = |\phi(\bar{x}_{k-1}) + \delta_k - \phi(\alpha)|$$

$$\leq |\phi(\bar{x}_{k-1}) - \phi(\alpha)| + |\delta|$$

e applicando il teorema del valor medio:

$$\leq |\phi'(\eta_k)| * |\bar{x}_{k-1} - \alpha| + \delta, \quad \eta_k - \alpha \leq \bar{x}_{k-1} - \alpha \leq \rho$$

con $|\phi'(\eta_k)| \leq \lambda$ per ipotesi del teorema.

$$|\bar{x}_k - \alpha| \leq \lambda * |\bar{x}_{k-1} - \alpha| + \delta$$

e applicando le ipotesi del teorema:

$$\leq \lambda(\sigma + \lambda^{k-1}(\rho - \sigma)) + \delta$$

e sostituendo l'espressione di $\delta = (1 - \lambda) * \sigma$

$$\leq \lambda * (\sigma + \lambda^{k-1}(\rho - \sigma)) + \sigma(1 - \lambda) \leq \lambda^k(\rho - \sigma) + \sigma$$

cvd

4.6 Radici di sistemi non lineari

4.6.1 Adattamento del metodo di Newton

Problema: data la funzione $f\colon \mathbb{R}^n \to \mathbb{R}^n$, trovare x_* tale che $f(x_*) = 0$.

Consideriamo un sottoinsieme $D \subset \mathbb{R}^n$ in cui f sia di classe C^1. Consideriamo la matrice jacobiana. Trovare gli zeri della funzione applicando il metodo di Newton equivale a risolvere l'equazione:

$$\begin{cases} f(x) = f(x_k) + J_f(x_k) * (x - x_k) \\ f(x) = 0 \end{cases}$$

Quindi

$$0 = f(x_k) + J_f(x_k) * (x_{k+1} - x_k)$$

$$(x_{k+1} - x_k) = -J_f^{-1}(x_k) * f(x_k) \quad \text{metodo di Newton}$$

Per evitare di invertire la matrice sul calcolatore, risolviamo il sistema lineare

$$J_f(x_k)(x_{k+1} - x_k) = -f(x_k)$$

Supponiamo di essere al k-esimo passo di Newton, allora poniamo

$$K = x_{k+1} - x_k$$

Risolviamo

$$J_f(x_k) * K = -f(x_k)$$

Poniamo

$$x_{k+1} = x_k + K$$

Come nel caso scalare, la convergenza del metodo di Newton è quadratica, se è sufficientemente vicino alla radice.

Lo jacobiano è non singolare se la radice è semplice.

4.6.2 Costo computazionale nella risoluzione del sistema

Pensiamo di risolvere il sistema lineare usando una fattorizzazione. Questo comporta un grande aumento del costo computazionale. Allora per risparmiare operazioni, si aggiorna lo jacobiano di f solo periodicamente, per sfruttare la fattorizzazione di $J_f(x_k)$ più volte. Questo rallenta la convergenza.

Pensando invece di usare metodi iterativi, consideriamo un ciclo di Newton con le iterazioni (*outer iteration*), e all'interno, nel passo in cui bisogna risolvere il sistema con la jacobiana, si inserisce un altro metodo iterativo (*inner iteration*).

Siccome l'obiettivo è quello di convergere alla radice, nelle prime iterazioni del metodo di Newton la soluzione non viene calcolata esattamente (si fissa un numero di iterazioni n_{max} basso per il metodo iterativo). Si è interessati a risolvere il sistema in modo più preciso tanto più si pensa di essere vicino alla radice.

5

Interpolazione polinomiale

5.1 Osservazioni introduttive

5.1.1 Introduzione del problema

Consideriamo $n+1$ coppie di dati, $(x_i, y_i)_{i=0}^n$, e fissiamo un modello di approssimazione. Abbiamo un set di funzioni $\phi_j(x)$ per $j = 1, \ldots, n$, che siano linearmente indipendenti.

Consideriamo la combinazione lineare

$$\sum_{j=0}^{n} \alpha_j \phi_j(x)$$

che restituisce 0 se e solo se $\alpha_j = 0 \forall j$.

Problema: dobbiamo determinare la funzione $\phi = \sum_{j=0}^n \alpha_j \phi_j(x)$, tale che $\phi(x_i) = y_i \forall i = 0, \ldots, n$.

Si può anche avere già f continua tale che $f(x_i) = y_i \forall i$, ma può essere conveniente approssimarla con polinomi.

Teorema 5.1 (teorema di Iston-Weierstrass)

Data una funzione f continua su $[a, b]$, allora per ogni $\varepsilon > 0$ esiste $n = n(\varepsilon)$ e un polinomio p_n tale che

$$|f - p_n| = \max_{x \in [a,b]} |f(x) - p_n(x)| \leq \varepsilon$$

Quindi, data una funzione continua, è una buona idea quella di usare polinomi per approssimarla.

5.1.2 Esistenza e unicità dell'interpolazione

Nella pratica, consideriamo $(x_i, y_i)_{i=0}^n$ e consideriamo p_n tale che $p_n(x_i) = y_i$ per $i = 0, \ldots, n$ (polinomio interpolante).

Teorema 5.2 (teorema di esistenza e unicità)

Dati $(x_i, y_i)_{i=1}^n$ nodi a due a due disgiunti, allora esiste ed è unico il polinomio $p_n(x)$ al più di grado n tale che $p_n(x_i) = y_i \forall i$.

Dimostrazione (dimostrazione teorica)

Scriviamo

$$p_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

e imponiamo le $n+1$ condizioni di interpolazione:

$$p_n(x_i) = \sum_{j=1}^{n} a_j (x_i)^j = y_i \forall i = 1, \ldots n$$

e abbiamo $n+1$ equazioni in $n+1$ incognite, equivalenti al sistema:

$$B\boldsymbol{a} = \boldsymbol{y}$$

dove $\boldsymbol{a} = (a_0, \ldots, a_n)^t$, $\boldsymbol{y} = (y_0, \ldots, y_n)$, e $B = B(x_0, \ldots, x_n)$ è una matrice con la seguente struttura:

$$\begin{matrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_2 & x_2^2 & \ldots & x_2^n \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{matrix}$$

e una matrice di questo tipo è detta di Vandermonde.

Si dimostra che

$$\det B(x_0, x_1, \ldots, x_n) = \prod_{i>j, i,j=0}^{n} (x_i - x_j) \quad \text{formula } *$$

e siccome i nodi sono a due a due distinti per ipotesi, il determinante è diverso da 0, ovvero esiste ed è unico il polinomio interpolante perché il sistema si può risolvere e tutti i coefficienti sono determinati.

Dimostrazione (dimostrazione della formula $*$)

Consideriamo la matrice di Vandermonde in cui al nodo x_0 sostituisco la generica variabile z, quindi

$$B(z, x_1, x_2, \ldots, x_n) = \begin{matrix} 1 & z & z^2 & \ldots & z^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{matrix}$$

Allora

$$\det B(z, x_1, \ldots, x_n)$$

è un polinomio nella variabile z di grado n per la formula di Laplace. Inoltre questo polinomio si annulla in $x_1, x_2, \ldots, x_n$ perché se a z sostituisco x_i, trovo una matrice con due righe uguali che ha determinante nullo. Allora si avrà:

$$p_n(z) = \det B(z, x_1, \ldots, x_n) = (z - x_1)(z - x_2) * \cdots * (z - x_n) * k(x_1, \ldots, x_n)$$

Valuto p_n in 0.

$$p_n(0) = (-1)^n * k * \prod_{i=0}^{n} x_i, \quad \text{relazione } \circ$$

Inoltre per definizione:

$$p_n(0) = \det B(0, x_1, \dots, x_n) = \det B(2:n, 2:n)$$

e sappiamo che

$$B(2:n, 2:n) = \operatorname{diag}(x_1, \dots, x_n) * \begin{matrix} 1 & x_1 & \dots & x_1^{n-1} \\ 1 & x_2 & \dots & x_2^{n-1} \\ \dots & \dots & \dots & \dots \\ 1 & x_n & \dots & x_n^{n-1} \end{matrix}$$

quindi

$$p_n(0) = \det B(x_1, \dots, x_n) * \prod_{i=0}^{n} x_i \text{formula } \circ\circ$$

ed eguagliando le due espressioni di $p_n(0)$:

$$(-1)^n * k(x_1, \dots, x_n) * \prod_{i=0}^{n} x_i = \det B(x_1, x_n) * \prod_{i=0}^{n} x_i$$

quindi per k ricavo l'espressione:

$$k = (-1)^n \det B(x_1, \dots, x_n)$$

Ipotesi induttiva: $\det B(x_1, \dots, x_n) = \prod_{i>j,i,j=1}^{n} (x_i - x_j)$ Allora

$$\det B(x_0, x_1, \dots, x_n) = \prod_{j=1}^{n} (x_0 - x_j) * k$$

$$= (-1)^n (x_1 - x_0) * (x_2 - x_0) * \dots * (x_n - x_0) * \det B(x_1, \dots, x_n)) * (-1)^n$$

$$= (x_1 - x_0) * (x_2 - x_0) * \dots * (x_n - x_0) * \prod_{j<i,j=1}^{n} (x_j - x_i)$$

e abbiamo dimostrato la formula $*$.

Nota: spesso la matrice di Vandermonde è mal condizionata, quindi nella pratica non si usa questo procedimento per ottenere i polinomi interpolanti.

Dimostrazione (dimostrazione costruttiva)

L'obiettivo è, come prima, quello di cercare un polinomio tale che $p_n(x_i) = y_i$, $\forall i = 0, \dots, n$ (condizione di interpolazione).

Premettiamo che i polinomi interpolanti sono unici, infatti, supponiamo per assurdo che esistano due polinomi $p_n, \bar{p}_n$ di grado n interpolanti, cioè che hanno come radici gli x_i, allora consideriamo $q_n = p_n - \bar{p}_n$, di grado n, che ha $n+1$ zeri. Infatti $q_n(x_i) = p_n(x_i) - \bar{p}_n(x_i) = y_i - y_i = 0$ allora q_n è identicamente nullo e quindi $p_n(x) = q_n(x)$.

Consideriamo una combinazione lineare di un opportuno set di funzioni (polinomi di Lagrange):

$$p_n(x) = \sum_{i=0}^{n} \alpha_i \phi_i(x)$$

dove $\alpha_i = y_i$ (sono le ordinate degli x_i) e $\phi_i = l_i(x)$ con $\lambda_i(x)$ polinomio di Lagrange.

$$l_i(x) = \prod_{j \neq i, j=0}^{n} \frac{x - x_j}{x_i - x_j}$$

Valutando $l_i(x)$ in un qualsiasi nodo x_k, segue che

1. se $x_k = x_i$, tutti i numeratori sono uguali ai denominatori e si ottiene $l_i(x) = 1$
2. se invece $x_k \neq x_i$, segue che $x_k = \bar{x}_j$, quindi un numeratore si annulla, e $l_i(x_k) = 0$.

In forma sintetica $l_i(x_k) = \delta_{ik}$.
Calcoliamo p_n in x_j:

$$p_n(x_j) = \sum_{i=0}^{n} y_i l_i(x_j) = \sum_{i=0}^{n} y_i \delta_{ij} = y_j$$

e abbiamo verificato la condizione di interpolazione per ogni $j = 0, \ldots, n$.

5.2 Altre scritture dei polinomi interpolanti

5.2.1 Scrittura dei polinomi interpolanti in funzione dei polinomi nodali

◊ **Definizione** Definiamo **polinomio nodale**

$$W_{n+1}(x) = \prod_{i=0}^{n} (x - x_i)$$

Cerchiamo una scrittura equivalente dei polinomi interpolanti:

$$\sum_{i=0}^{n} y_i l_i =$$

$$= \sum_{i=0}^{n} y_i \frac{\prod_{i \neq j, j=0}^{n} (x - x_j)}{\prod_{i \neq j, j=0}^{n} (x_i - x_j)}$$

e tenendo conto dell'espressione di $w_{n+1}(x)$:

$$= \sum_{i=0}^{n} y_i \frac{w_{n+1}(x)}{x - x_i} * \frac{1}{\prod_{j \neq i, j=1}^{n} (x_i - x_j)}$$

Osserviamo che:

$$w_{n+1}(x)' = (\prod_{j=0}^{n} (x - x_j))' = \sum_{i=0}^{n} \prod_{j \neq i, j=0}^{n} (x - x_j)$$

e valutando in x_k:

$$w_{n+1}(x_k)' = \prod_{j \neq k, j=0}^{n} (x_k - x_j)$$

quindi, tornando all'espressione del polinomio interpolante:

$$\sum_i y_i l_i(x) =$$

$$= W_{n+1}(x) * \sum_{i=0}^{n} y_i \frac{1}{x - x_i} * \frac{1}{w'_{n+1}(x_i)}$$

5.2.2 Considerazioni sui costi computazionali

Poniamo

$$p_n(x) = w_{n+1}(x) * \sum_{i=0}^{n} \frac{z_i}{x - x_i}$$

dove $z_i = \frac{y_i}{\prod_{i \neq j, j=0}^{n} (x_i - x_j)}$.

Ci si chiede qual è il costo computazionale della valutazione di $p_n(x)$ in un x assegnato. Sono necessarie:

1. n sottrazioni per calcolare $x - x_i$ (al denominatore);
2. $n^2/2$ sottrazioni per calcolare gli $x_i - x_j$ nella produttoria, e questo equivale a calcolare le entrate della matrice $L = (l_{ij})$ con 0 sulla diagonale principale, tale che $l_{ij} = -l_{ji}$;
3. n^2 operazioni per calcolare gli z_i, in particolare per calcolare ogni z_i, una volta che gli $x_i - x_j$ sono noti, sono necessari solo n prodotti. Siccome gli z_i sono n abbiamo in totale n^2 operazioni;
4. n addizioni (defivano dalla sommatoria) e n moltiplicazioni per calcolare $\frac{z_i}{x - x_i}$, divisione tra due numeri già noti;
5. n operazioni per la valutazione del polinomio nodale di cui già conosco i fattori.

$$tot = n^2/2 + n^2$$

più una quantità trascurabile dell'ordine di n operazioni.

Ci si chiede quanto costa calcolare il polinomio interpolante in un nuovo punto $\bar{x}$, dopo aver effettuato il calcolo descritto sopra. Le operazioni riferite ai punti 2 e 3 (calcolo completo degli z_i) sono indipendenti dal punto in cui si valuta il polinomio, e non vanno ripetute a ogni passaggio. Si hanno quindi solo $2n$ operazioni additive e $2n$ moltiplicative. Rispetto alla risoluzione del sistema delle $n + 1$ equazioni nelle $n + 1$ incognite (come nella dimostrazione 1), dove la fattorizzazione LU richiede n^3 operazioni, si risparmiano operazioni.

Supponiamo di aggiungere un nuovo punto (x_{n+1}, y_{n+1}) ai dati: si vuole calcolare un nuovo polinomio interpolante di grado $n + 1$.

Gli z_i sono già stati calcolati, si ha che il nuovo polinomio è:

$$p_{n+1}(x) = w_{n+2}(x) * [\sum_{i=0}^{n} (\frac{z_i^{(n+1)}}{x - x_i}) + \frac{z_{n+1}^{(n+1)}}{x - x_{n+1}}]$$

$z_i^{(n+1)} = \frac{y_i}{\prod_{j=0}^{n}(x_i - x_j)} * \frac{1}{x_i - x_{n+1}}$, quindi

$$z_i^{(n+1)} = \frac{z_i^{(n)}}{x_i - x_{n+1}}$$

Inoltre

$$w_{n+2}(x) = w_{n+1}(x) * (x - x_{n+1})$$

e si ricava che

$$p_{n+1}(x) = w_{n+1}(x) * (x - x_{n+1}) * [\sum_{i=0}^{n}(\frac{z_i^{(n)}}{(x - x_i)(x_i - x_{n+1})}) + \frac{z_{n+1}^{(n+1)}}{x - x_{n+1}}]$$

quindi le operazioni totali in più, avendo già calcolato $p_n(x)$, sono una sottrazione per $x - x_{n+1}$, n sottrazioni per $x_i - x_{n+1}$, n moltiplicazioni per $z_i^{(n+1)}$, n addizioni per la sommatoria, n moltiplicazioni per $z_i^n/(x - x_i)$, 2 operazioni per l'ultimo termine.

In totale abbiamo $2n$ addizioni e $2n$ moltiplicazioni.

5.2.3 Forma di Newton

Il polinomio interpolante è unico ma ci sono diverse riscritture. La forma di Newton permette di calcolare i polinomi interpolanti aumentando le operazioni additive e diminuendo quelle moltiplicative.

Il polinomio interpolante

$$p_n(x) = \sum_{k=0}^{n} a_k * \phi_k(x)$$

viene riscritto in funzione della base dei polinomi nodali, ponendo

$$\phi_k = \prod_{j=1}^{k}(x - x_j)$$

Gli a_k vengono posti uguali alla differenza divisa di ordine k.

◊ **Definizione** Si indica con $f[x_0, \ldots, x_k]$ la differenza divisa di ordine k ($k + 1$ nodi). La differenza divisa si definisce induttivamente, ponendo:

$$f[x_0] = f(x_0) \text{per definizione}$$

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad \text{rapporto incrementale}$$

$$f[x_0, \ldots, x_k] = \frac{f[x_1, \ldots, x_k] - f[x_0, \ldots, x_{k-1}]}{x_k - x_0}$$

Vogliamo dimostrare che

$$p_n(x) = f[x_0] + f[x_0, x_1] * (x - x_0) + f[x_0, x_1, x_2] * (x - x_0) * (x - x_1) + \cdots + f[x_0, x_1, \ldots, x_n) * (x - x_0) * (x - x_1) * (x - x_{n-1})$$

è un polinomio interpolante, dimostriamo prima il lemma di Neville.

Lemma 5.1 (formula di Neville)

Dato $I = \{i_1, \ldots, i_k\}$ un insieme di indici, e definendo $p(i_0, \ldots, i_k)$ il polinomio valutato nei nodi (x_{i_j}, y_{i_j}) per $j = 0, \ldots, n$, si ha che:

$$p_k(x) = \frac{(x - x_{i_0}) * p_{k-1}(x)(i_1, \ldots, i_k) - (x - x_{i_k}) * P_{k-1}(x)(i_0, \ldots, i_{k-1})}{x_{i_k} - x_{i_0}}$$

è un polinomio interpolante.

Dim. Il lemma dà una rappresentazione ricorsiva del polinomio interpolante.

Per calcolare l'espressione di $p(i_0)$ si tiene conto che il termine $(x - x_{i_0}) * p_{k-1}(i_1, \ldots, i_k)$ è nullo, quindi rimane

$$p(i_0) = \frac{-(x - x_{i_k}) * P_{k-1}(x_{i_0})(i_0, \ldots, i_{k-1})}{x_{i_k} - x_{i_0}} = p_{k-1}(x_{i_0})(i_0, \ldots, i_{k-1}) = y_{i_0}$$

Valutando invece $p(i_k)$ si annulla il secondo termine, e rimane:

$$p(i_k) = \frac{(x_{i_k} - x_{i_0}) * p_{k-1}(i_1, \ldots, i_k)(x_{i_k})}{x_{i_k} - x_{i_0}} =$$

$$p_{k-1}(i_1, \ldots, i_k)(x_{i_k}) = y_{i_k}$$

per un generico $j = 1, 2, \ldots, k-1$, si ha:

$$p(i_j) = p(x_{i_j}) = \frac{(x_{i_j} - x_{i_0}) * p_{k-1}(i_1, \ldots, i_k)(x_{i_j}) - (x_{i_j} - x_{i_k}) * p_{k-1}(i_0, \ldots, i_{k-1})(x_{i_j})}{x_{i_k} - x_{i_0}}$$

$$= \frac{(x_{i_j} - x_{i_0}) * y_{i_j} - (x_{i_j} - x_{i_k}) * y_{i_j}}{x_{i_k} - x_{i_0}}$$

$$= y_{i_j}$$

Abbiamo quindi verificato tutte le condizioni di interpolazione.

Teorema 5.3 (dimostrazione della forma di Newton)

$$p_n(x) = f[x_0] + f[x_0, x_1] * (x - x_0) + f[x_0, x_1, x_2] * (x - x_0) * (x - x_1) + \cdots + f[x_0, x_1, \ldots, x_n] * (x - x_0) * \cdots * (x - x_{n-1})$$

è un polinomio interpolante.

Dimostrazione

Dimostriamo l'asserto per induzione.

1. Se $n = 0$, otteniamo che $p_0(x) = f[x_0] = y_0$, per definizione.
2. Per ipotesi induttiva, assumo che sia vero che il polinomio di Newton è il polinomio interpolante per il grado $n-1$ rispetto alla base nodale, e lo dimostro per n.
3. Supponiamo che $p_n(x)$ sia il polinomio interpolante di grado n rispetto alla base delle $\phi_k(x)$ date dai polinomi nodali di indice k, allora possiamo scriverlo come:

$$p_n(x) = p_{n-1}(x) + a_n * (x - x_0)(x - x_1) * \cdots * (x - x_n)$$

vogliamo dimostrare che $a_n = f[x_0, \ldots, x_n]$. Consideriamo la formula di interpolazione di Neville. Allora

$$p_n(x) = \frac{(x - x_0) * p_{n-1}(1, \ldots, n) - (x - x_n) * p_{n-1}(x)(0, \ldots, n-1)}{x_n - x_0}$$

Dall'ipotesi di induzione segue che: $p_{n-1}(1, \ldots, n)$ è il polinomio interpolante di grado $n-1$ sui nodi $x_1, \ldots, x_n$, e per ipotesi di induzione il coefficiente K_1 del termine di grado massimo è $f[x_1, \ldots, x_n]$. $p_{n-1}(0, \ldots, n-1)$ è il polinomio interpolante rispetto ai nodi $(x_0, \ldots, x_{n-1})$ e il coefficiente K_2 del suo termine di grado massimo è $f[x_0, \ldots, x_{n-1}]$. Quindi, nel polinomio p_n, in base alla formula di Neville e a quanto osservato, il coefficiente del termine di grado massimo è:

$$K = \frac{1 * K_1 - 1 * K_2}{x_k - x_0}$$

$$= \frac{1 * f[x_1, \ldots, x_n] - 1 * f[x_0, \ldots, x_{n-1}]}{x_n - x_0} = f[x_0, \ldots, x_n] \quad \text{per definizione}$$

Osservazione 5.1

La differenza divisa non dipende dall'ordine dei punti. Infatti, confrontando la scrittura del polinomio interpolante in forma di Newton e di Lagrange, per il coefficiente a_n si ha:

$$a_n = f[x_0, \ldots, x_n] = \sum_{i=0}^{n} \frac{f(x_i)}{p'_{n+1}(x_i)}$$

siccome nella sommatoria non c'è dipendenza dall'ordine dei punti, anche la differenza divisa non ne dipende in aritmetica esatta.

5.2.4 Algoritmo alla Neville

Consideriamo i due vettori colonna $x = (x_0, x_1, \ldots, x_n)$ e $y = (f_0, f_1, \ldots, f_n)$ dove $f_i = f(x_i)$. L'algoritmo alla Neville porta alla costruzione di una matrice triangolare inferiore A, dove le entrate sono determinate nel seguente modo:

for(i=1:n-1)

for(j=n:i)

$Y(j) = (y(j) - y(j-1))/(x(j) - x(j-i))$ differenza divisa

end

end

Ad esempio, presi 6 punti $(x_0, x_1, x_2, x_3, x_4, x_5$ la matrice che risulta dall'algoritmo di Neville è:

$$
\begin{array}{cccccccc}
x_0 & f(x_0) & & & & & & \\
x_1 & f(x_1) & f[x_1,x_0] & 0 & 0 & 0 & 0 \\
x_2 & f(x_2) & f[x_2,x_1] & f[x_2,x_1,x_0] & 0 & 0 & 0 \\
x_3 & f(x_3) & f[x_3,x_2] & f[x_3,x_2,x_1] & f[x_3,\ldots,x_0] & 0 & 0 \\
x_4 & f(x_4) & f[x_4,x_3] & f[x_4,x_3,x_2] & f[x_4,\ldots,x_1] & f[x_4,\ldots,x_0] & 0 \\
x_5 & f(x_5) & f[x_5,x_4] & f[x_5,x_4,x_3] & f[x_5,\ldots,x_2] & f[x_5,\ldots,x_1] & f[x_5,x_0]
\end{array}
$$

Il ciclo viene fatto a ritroso perché, calcolando la triangolare inferiore con la colonna partendo dal basso, si possono sovrascrivere direttamente i vettori. In questo algoritmo, l'entrata A_{ij} diventa la differenza divisa tra gli elementi $A_{i,j-1}$ e $A_{i-1,j-1}$, per ogni j.

Numero di operazioni effettuate: per ogni elemento faccio una divisione e due sottrazioni, e la triangolare inferiore ha $n^2/2$ elementi, quindi vengono effettuate $2 * n^2/2 = n^2$ operazioni additive e $n^2/2$ operazioni moltiplicative.

5.2.5 Algoritmo di Hornern

Per la valutazione del polinomio di Newton in un punto x si usa l'algoritmo di Horner: ad esempio, nel caso di un polinomio di grado 3, il polinomio di Newton si rappresenta come:

$$p_n = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_0)(x - x_1)(x - x_2)$$

Raccogliamo $x - x_0$ tra gli ultimi tre addendi:

$$p_n = a_0 + (x - x_0) * [a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2)]$$

Poi raccogliamo $x - x_1$ tra gli ultimi due addendi:

$$p_n = a_0 + (x - x_0) * [a_1 + (x - x_1) * [a_2 + (x - x_2) * a_3]]$$

e per eseguire le operazioni partiamo dall'ultima parentesi. Tuttavia le operazioni necessarie per la valutazione del polinomio in un punto sono di più rispetto a quelle del polinomio di Lagrange.

5.3 Stabilità condizionamento e convergenza

5.3.1 Errore analitico

Teorema 5.4

Sia I_x il più piccolo intervallo che contiene i nodi di interpolazione, e supponiamo che contenga anche il punto x in cui vogliamo valutare l'errore. Supponiamo che $F \in C_{n+1}(I_x)$. Allora esiste $\xi \in I_x$ tale che

$$E_x = |f(x) - p_n(x)| = \frac{f^{(n+1)}(\xi)}{(n+1)!} * w_{n+1}(x)$$

(analogia con la formula di Taylor, centrata però in tutti gli x_i)

Dimostrazione

Se x è un nodo di interpolazione, allora $E_0 = 0$.

Altrimenti, supponiamo di considerare $t \in I_x$, $t \neq x$. Introduciamo la funzione ausiliaria

$$g(t) = E(t) - \frac{w_{n+1}(t)}{w_{n+1}(x)} * E(x)$$

di classe $C_{n+1}(I_x)$.

Osserviamo che valutando g in uno dei nodi di interpolazione x_i, siccome $w_{n+1}(x_i) = 0$ e $E(x_i) = 0$ si ha:

$$g(x_i) = E(x_i) - \frac{w_{n+1}(x_i)}{w_{n+1}(x)} * E(x) = 0$$

e $g(x_i) = 0 \forall i = 1, \ldots, n$.

Valutando g nel nodo x si ha:

$$g(x) = E(x) - \frac{w_{n+1}(x)}{w_{n+1}(x)} * E(x) = E(x) - E(x) = 0$$

quindi la funzione si annulla in $n+2$ nodi, gli x_i e v.

Per il teorema di Rolle: $g'(t)$ si annulla in $n+1$ punti, e $g^{(n+1)}(t)$ si annulla in un punto $\xi \in I_x$.

$$g^{(n+1)}(\xi) = E^{(n+1)}(\xi) - \frac{w_{n+1}^{(n+1)}(t)}{w_{n+1}(x)} * E(x)$$

$$= f^{(n+1)}(\xi) - \frac{n!}{w_{n+1}(x)} * E(x)$$

e siccome $g^{(n+1)}(\xi) = 0$ per il teorema di Rolle, si ha

$$E(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} * w_{n+1}(x)$$

Rappresentiamo I_x usando le differenze divise. Chiamiamo $E_n(f)(x)$ l'errore dell'approssimazione di $f(x)$ con P_n. Sia $P_n(t)$ il polinomio interpolante. Sia $x \neq x_i$, considerato come nuovo nodo di interpolazione.

Scriviamo il polinomio interpolante nei nodi x_i, con $i = 0, \ldots, n+1$. Aumento di 1 il grado del polinomio interpolante

$$p_{n+1}(t) = p_n(t) + f[x_0, \ldots, x_n, x] * (t - x_0) * (t - x_1) * \cdots * (t - x_n)$$

Siccome $f(x) = p_{n+1}(x)$ per la formula di Newton, ricavo che

$$E_n(f)(x) = f(x) - p_n(x) = f[x_0, \ldots, x_n, x] * (x - x_0) * (x - x_1) * \cdots * (x - x_n)$$

e se $f \in C_{n+1}(I_x)$, segue che $f[x_0, \ldots, x_n, x] = \frac{f^{(n+1)}(\xi)}{(n+1)!}$.

5.3.2 Convergenza del polinomio interpolante

Sia $f \in C^0(a,b)$, allora possiamo definire la norma infinito di f come $\max \max_{[a,b]} |f(x)|$. Vogliamo studiare l'errore di interpolazione E_n per $n \to \infty$.

Introduco una matrice di interpolazione sull'intervallo $[a,b]$, cioè definisco $X = (x_{ij})$ di dimensione infinita che contiene i nodi di interpolazione, è una matrice triangolare inferiore, che su ogni riga ha i nodi di interpolazione del polinomio che ha come grado l'indice della riga.

Fissati f, X, chiamiamo $E_n(f,X) = |f - p_n|$.

Confronto E_n con l'errore di migliore approssimazione $E_n^*(f)$, cioè $E_n^*(f) = |f - P_n^*|$ dove P_n^* è tale che $|f - P_n^*| \leq |f - Q|$ per ogni Q polinomio di grado n.

P_n^* non passa necessariamente dai nodi. Sappiamo per il teorema di Iston-Weierstrass che $E_n \to 0$ per $n \to \infty$.

Esiste una relazione tra errore di interpolazione e errore di miglior approssimazione.

Supponiamo di avere una funzione continua su $[a,b]$, e una matrice di interpolazione i su tale intervallo. Vale il seguente risultato.

Teorema 5.5

L'errore di interpolazione

$$E_n(f,X) \leq E_n^*(f) * (1 + \Lambda_{n,X})$$

dove

$$\Lambda = |\sum_{j=0}^{n} |L_{j,n}||$$

dove $L_{j,n}$ è il j-esimo polinomio di Lagrange di grado n (Λ viene chiamata costante di Lebesgue).

Dimostrazione

Poniamo $P_n = P_n(X,f)$, per ogni $y \in [a,b]$, consideriamo

$$|E_y| = |f(y) - P_n(y)| = |f(y) - P_n(y) + P_n^*(y) - P_n^*(y)|$$

$$\leq |f(y) - P_n^*(y)| + |P_n^*(y) - P_n(y)|$$

La differenza tra il polinomio di miglior approssimazione e la funzione può essere maggiorata con E_n^*, quindi

$$\leq E_n^*(f) + |P_n^*(y) - P_n(y)|$$

Argomento di proiezione: $P_n^*(f)(y) = P_n(P_n^*(f))(y)$ (infatti il polinomio interpolante di un polinomio è il polinomio stesso), e sostituendo:

$$\leq E_n^*(f) + |P_n(P_n^*(y)) - P_n(y)|$$

Argomento di linearità: la differenza tra polinomi interpolanti di H_1 e H_2 è il polinomio interpolante della differenza $H_1 - H_2$, quindi

$$\leq E_n^*(f) + |P_n(P_n^* - f)(y)|$$

e scrivendo il polinomio interpolante in forma di Lagrange:

$$= E_n^*(f) + |\sum_{j=0}^{n}(f - P_n^*)(x_j^n)L_{j,n}(y)|$$

con la disuguaglianza triangolare

$$\leq E_n^*(f) + \sum j = 0^n |f - p_n^*(f)|(x_j^n) * |L_{j,n}|(y)$$

e maggiorando $(P_n^* - f)(x_j^n)$ con $E_n^*(f)$ che è il massimo sull'intervallo:

$$\leq E_n^*(f) + \sum j = 0^n E_n^*(f) * |L_{j,n}|(y)$$

e portando E_n fuori dalla sommatoria

$$\leq E_n^*(f) * (1 + \sum j = 0^n |L_{j,n}|(y))$$

$$\leq E_n * (1 + \lambda_n)$$

dove Λ è come definito sopra.

5.3.3 Nodi di Chebyshev

Esiste una matrice X^* che minimizza la costante di Lebesgue $\Lambda_n(x)$, ma non ne abbiamo la forma esplicita. Si considerano i nodi del polinomio di Chebyshev, che danno la migliore approssimazione di X^* a meno di termini di ordine inferiore.

$\Diamond$ **Definizione** Chiamiamo **polinomio di Chebyshev di grado** m, $P_m(x)$ il polinomio definito come $\cos(m * \arccos x)$.

Poniamo $\theta = \arccos x$, allora $P_m(\theta) = \cos(m\theta)$, e dobbiamo risolvere $\cos(m\theta) = 0$, e questo è vero se

$$m\theta = \pi/2 + k\pi, k \in \mathbb{Z}$$

quindi gli zeri sono

$$\theta_k = \frac{\pi}{2m} + \frac{k\pi}{m}$$

Definiamo

$$x_k = \cos\theta_k,\ k = 1, \ldots, m-1$$

$x_k \in [-1, 1]$, sono tutti interni all'intervallo, e hanno la proprietà che si infittiscono verso gli estremi e sono più radi al centro.

$x \in [-1, 1]$, consideriamo $z = \alpha x + \beta$, impongo $a = -\alpha + \beta$, $b = \alpha + \beta$, e ricaviamo che $\alpha = (b - a)/2$ e $\beta = (b + a)/2$, quindi i nodi mappati su (a, b) sono $(b + a)/2 + x_k(b - a)/2$. In generale, per ogni scelta dei nodi di interpolazione, esiste una costante $c > 0$ tale che

$\Lambda_n(x) > 2/\pi \log(+1) - c$, e quindi $\lambda_n(x) \to \infty$ per $n \to \infty$.

Teorema 5.6 (teorema di Faber)

Data X matrice di interpolazione sull'intervallo $[a, b]$, esiste almeno una funzione f continua su $[a, b]$ tale che $P_n(X, f)$ non converge uniformemente a f per $n \to \infty$.

Questo significa che una matrice di interpolazione non va bene per tutte le funzioni.

5.3.4 Funzione di Runge

Consideriamo la funzione di Runge:

$$f(x) = \frac{1}{1 + x^2}$$

L'interpolazione polinomiale di questa funzione con nodi equispaziati non dà buoni risultati. Invece con i nodi di Chebyshev si ottiene un risultato migliore.

5.3.5 Condizionamento

Supponiamo che $\bar{f}(x_i)$ siano le perturbazioni di $f(x_i)$, chiamiamo $\bar{p}_n(x)$ il polinomio interpolante rispetto ai valori perturbati. Vale che

$$|p_n(x) - \bar{p}_n(x)| = \max_{x \in [a,b]} |\sum_{i=0}^{n} (f(x_i) - \bar{f}(x_i)) * L_i(x)|$$

e per la disuguaglianza triangolare

$$= \max \sum_i |f(x_i) - \bar{f}(x_i)| * |L_i(x)|$$

$$\leq \sum_{i=0}^{n} |f(x_i) - \bar{f}(x_i)| * \max_{x \in [a,b]} \sum_{i=0}^{n} |L_i(x)|$$

$$= \sum_{i=0}^{n} |f(x_i) - \bar{f}(x_i)| * \Lambda(X)$$

quindi la costante di Lebesgue può essere interpretata come numero di condizionamento e amplifica la perturbazione sui dati.

$\Lambda_n(x)$ cresce all'aumentare di n, e vale il seguente risultato: se consideriamo nodi equispaziati, $\Lambda_n = \frac{2^{n+1}}{E_n \log n}$ e si ha una crescita esponenziale. Invece, considerando i nodi di Cevicev, la crescita è solo logaritmica.

5.4 Interpolazione polinomiale a tratti (spline)

5.4.1 Definizione di *spline*

◊ **Definizione** Consideriamo una funzione $s(x) \colon [a, b] \to \mathbb{R}$, diciamo che $s(x)$ è una ***spline* di grado** k se, dato un intervallo $[a, b]$ su cui consideriamo il partizionamento

$$\{a = x_0 < x_1 < \cdots < x_n = b\}$$

s è definita come

$$s(x) = \begin{cases} s_0(x) & \iff x \in [x_0, x_1] \\ s_1(x) & \iff x \in [x_1, x_2] \\ s_{n-1}(x) & \iff x \in [x_{n-1}, x_n] \end{cases}$$

dove le s_i sono polinomi di grado k per $i = 0, \ldots, n-1$, localmente C^∞, e tali che globalmente $s_x \in C_{k-1}$.

Nel caso $k = 1$, consideriamo l'intervallo $[a, b]$ e una partizione, in ciascun sottointervallo si ha un polinomio di grado 1, e le s_i sono localmente polinomi di grado 1 e s è di classe C^0 sull'intervallo; si ha l'interpolazione lineare a tratti.

Nel caso $k = 3$ si ha l'interpolazione *spline* cubica, si considera una partizione di $[a, b]$, in ciascun sottointervallo definisco un polinomio di grado 3. s è di classe C^2 sull'intervallo.

5.4.2 Sistema per la determinazione della *spline*

$s_i(x)$ sono polinomi di grado 3, con $i = 0, \ldots, n-1$. Abbiamo $4n$ coefficienti da determinare.

Dobbiamo imporre le seguenti condizioni:

- $n+1$ condizioni di interpolazione, perché la *spline* deve valere y_i in x_i;
- $n-1$ condizioni di raccordo con continuità, cioè $s_{i-1}(x_i) = s_i(x_i)$ per $i = 1, \ldots n-1$;
- $2n-2$ condizioni sulle derivate, cioè $s'_{i-1}(x_i) = s'_i(x_i)$ e $s''_{i-1}(x_i) = s''_i(x_i)$ perché s dev'essere di classe C^2.

Quindi mancano due condizioni per avere un sistema determinato di $4n$ equazioni in $4n$ incognite.

In base alle due condizioni aggiuntive che si impongono si hanno diversi tipi di *spline*:

***Spline* naturale** imponiamo che $s(a)$=$s(b) = 0$.
***Spline* completa e vincolata** $s'_0(a) = f'(a)$ e $s'_{n-1}(b) = f'(b)$.
***Spline* periodica** supponiamo $T = b - a$, allora $s'_0(a) = s'_{n-1}(b)$ e analogamente $s''_0(a) = s''_{n-1}(b)$.

Abbiamo un sistema di $4n$ equazioni in $4n$ incognite, che però ha grandi dimensioni ed è spesso mal condizionato.

5.4.3 Metodo dei momenti

Definizione della funzione: sia M_i la derivata seconda di s in x_i, per $i = 1, \ldots, n$ (incognite da calcolare), dove M_i sono detti momenti. Siccome s_i ha grado 3, la sua derivata è un polinomio di grado 1, inoltre, imponendo le condizioni:

$$\begin{cases} s''_i(x_i) = M_i \\ s''_{i+1}(x_{i+1}) = M_{i+1} \end{cases}$$

si ottiene

$$s''_i(x) = M_{i+1}(x - x_i)/h_i - M_i(x - x_{i+1})/h_i$$

dove $h_i = x_{i+1} - x_i$.

Integrando due volte otteniamo:

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

Integrando nuovamente

$$s_i(x) = M_{i+1}\frac{(x-x_i)^3}{6H_i} - M_i\frac{(x-x_{i+1})^3}{6H_i} + \alpha * (x-x_i) + \beta_i \quad \text{equazione } *$$

Imposizione delle condizioni di interpolazione: per determinare i nuovi parametri α_i, β_i, imponiamo le condizioni di interpolazione e otteniamo:

$$s_i(x_i) = -M_i\frac{(x_i-x_{i-1})^3}{6H_i} + \beta_i = f(x_i)$$

Inoltre

$$s_i(x_{i+1}) = M_{i+1} * (x_{i+1}-x_i)^3/(6H_i) + \alpha_i(x_{i+1}-x_i) + \beta_i = f(x_{i+1})$$

dove l'ultimo passaggio vale per la condizione di raccordo con continuità $s_i(x_{i+1} = s_{i+1}(x_{i+1})$. Quindi, per $i = 0, \ldots, n-1$ otteniamo:

$$\begin{cases} M_i\frac{H_i^2}{6} + \beta_i = f(x_i) \\ M_{i+1}\frac{H_i^2}{6} + \alpha_i H_i + \beta_i = f(x_{i+1}) \end{cases}$$

Risolviamo il sistema:

$$\begin{cases} M_i\frac{H_i^2}{6} - f(x_i) = -\beta_i \\ M_{i+1}\frac{H_i^2}{6} + \alpha_i H_i - f(x_{i+1}) = -\beta_i \end{cases}$$

ed eguagliando le due equazioni otteniamo:

$$M_i\frac{H_i^2}{6} - f(x_i) = M_{i+1}\frac{H_i^2}{6} + \alpha_i H_i - f(x_{i+1})$$

$$f(x_{i+1}) - f(x_i) + (M_i - M_{i+1})\frac{H_i^2}{6} = \alpha_i H_i$$

$$\alpha_i = \frac{f(x_{i+1}) - f(x_i)}{H_i} + (M_i - M_{i+1})\frac{H_i}{6}$$

invece ricaviamo β_i dalla prima equazione:

$$\beta_i = -M_i\frac{H_i^2}{6} + f(x_i)$$

e come soluzione del sistema otteniamo

$$\begin{cases} \beta_i = f(x_i) - M_i H_i^2/6 \\ \alpha_i = \frac{f(x_{i+1})-f(x_i)}{H_i} - H_i/6(M_{i+1} - M_i) \end{cases}$$

Imposizione delle condizioni di raccordo con continuità delle derivate:

$$s'_{i-1}(x) = \frac{M_i(x-x_{i-1})^2}{2h_{i-1}} - M_{i-1}\frac{(x-x_i)^2}{2H_{i-1}} + \alpha_i$$

$$= \frac{M_i(x-x_{i-1})^2}{2h_{i-1}} - M_{i-1}\frac{(x-x_i)^2}{2H_{i-1}} + \frac{f(x_i)-f(x_{i-1})}{H_{i-1}} - H_{i-1}/6(M_i - M_{i-1})$$

Quindi, valutando in x_i si ottiene:

$$s'_{i-1}(x_i) = \frac{M_i h_{i-1}}{2} + \frac{f(x_i)-f(x_{i-1})}{H_{i-1}} - H_{i-1}/6(M_i - M_{i-1})$$

analogamente

$$s'_i(x_i) = \frac{M_{i+1} h_i}{2} + \frac{f(x_{i+1})-f(x_i)}{H_i} - H_i/6(M_{i+1} - M_i)$$

ed eguagliando le due espressioni:

$$M_{i-1} * H_{i-1}/6 + M_i(H_{i-1}+H_i)/3 + M_{i+1}H_i/6 = \frac{f(x_{i+1})-f(x_i)}{H_i} - \frac{f(x_i)-f(x_{i-1})}{H_{i-1}}$$

e inserendo l'espressione della differenza divisa:

$$M_{i-1} * H_{i-1}/6 + M_i(H_{i-1}+H_i)/3 + M_{i+1}H_i/6 = f[x_{i+1}, x_i, x_{i-1}] * (H_i - H_{i-1})$$

e dividendo ambo i membri per $H_i - H_{i-1}$ otteniamo:

$$M_{i-1} * \frac{H_{i-1}}{6(h_i - h_{i-1})} + M_i/3 + M_{i+1}\frac{H_i}{6(h_i - h_{i-1})} = f[x_{i+1}, x_i, x_{i-1}]$$

e moltiplicando per 6

$$M_{i-1} * \frac{H_{i-1}}{H_i - H_{i-1}} + 2M_i + M_{i+1}\frac{H_i}{H_i - H_{i-1}} = 6f[x_{i+1}, x_i, x_{i-1}]$$

Chiamiamo

$$\gamma_i = \frac{H_{i-1}}{H_i - H_{i-1}}$$

e simmetricamente

$$\delta_i = \frac{H_i}{H_i - H_{i-1}}$$

e sono due quantità positive, che hanno la proprietà

$$\gamma_i + \delta_i = 1, \ \forall i$$

per qualsiasi tipo di nodi.

Abbiamo $n-1$ equazioni in $n+1$ incognite, date da:

$$M_{i-1} * \gamma_i + 2M_i + M_{i+1}\delta_i = 6f[x_{i+1}, x_i, x_{i-1}]$$

Imposizione delle condizioni iniziali: la situazione dipende dalle condizioni imposte.

Spline naturale: imponiamo che $M_0 = s''(x_0) = 0$, e $M_n = s''(x_n) = 0$. Allora otteniamo un sistema $A\boldsymbol{x} = \boldsymbol{b}$ dove A è una matrice tridiagonale della forma:

$$A = \begin{matrix} 2 & \delta_1 & 0 & 0 & 0 \\ \gamma_2 & 2 & \delta_2 & 0 & 0 \\ 0 & \gamma_3 & 2 & \dots & 0 \\ 0 & 0 & \dots & 2 & \delta_{i-1} \\ 0 & 0 & 0 & \gamma_{n-1} & 2 \end{matrix}$$

$$\boldsymbol{x} = \begin{matrix} M_1 \\ M_2 \\ \dots \\ M_{n-2} M_{n-1} \end{matrix}$$

è il vettore delle incognite, mentre il termine noto è

$$\boldsymbol{x} = \begin{matrix} f[x_0, x_1, x_2] \\ f[x_1, x_2, x_3] \\ f[x_2, x_3, x_4] \\ \dots \\ f[x_{n-2}, x_{n-1}, x_n] \end{matrix}$$

La matrice è diagonalmente dominante sulle righe.

Se i nodi sono equispaziati, cioè se $H_i = H \forall i$, la matrice è simmetrica. Per risolvere il sistema si può usare l'eliminazione gaussiana senza pivot, oppure metodi diretti che convergono per la dominanza diagonale stretta.

Spline completa: imponiamo le condizioni

$$s'(x_0) = f'(a), \quad s'(x_n) = f'(b)$$

Tenendo conto che

$$s'(x) = \frac{M_i(x - x_{i-1})^2}{2h_{i-1}} - M_{i-1}\frac{(x - x_i)^2}{2H_{i-1}} + \frac{f(x_i) - f(x_{i-1})}{H_{i-1}} - H_{i-1}/6(M_i - M_{i-1})$$

si ha

$$s'(x_0) = -M_0 H_0/2 + f[x_0, x_1] + H_0/6(M_0 - M_1)$$

$$f'(x_0) = H_0 f[x_0, x_0] = s'(x_0) = -M_0 H_0/3 - M_1 H_0/6 + H_0 f[x_0, x_1]$$

che si riscrive come

$$H_0/3M_0 - H_0/6M_1 = (f[x_0, x_1] - f[x_0, x_0]) * H_0 = f[x_0, x_0, x_1] * H_0$$

dividendo per H_0 e moltiplicando per 6:

$$2M_0 + M_1 = 6f[x_0, x_0, x_1]$$

Analogamente imponendo $f'(x_n) = s'(x_n)$ si ottene:

$$M_{n-1} + 2M_n = 6f[x_{n-1}, x_n, x_n]$$

In questo caso abbiamo un sistema di $n+1$ equazioni della forma $A\boldsymbol{x} = \boldsymbol{b}$ con:

$$A = \begin{matrix} 2 & 1 & 0 & 0 & 0 \\ \gamma_1 & 2 & \delta_1 & 0 & 0 \\ 0 & \gamma_2 & 2 & \delta_2 & 0 \\ 0 & 0 & \gamma_{n-1} & 2 & \delta_{n-1} \\ 0 & 0 & 0 & 1 & 2 \end{matrix}$$

Il vettore delle incognite è

$$\begin{matrix} M_0 \\ M_1 \\ \dots \\ M_{n-1} \\ M_n \end{matrix}$$

Il termine noto è:

$$6 * \begin{matrix} f[x_0, x_0, x_1] \\ f[x_0, x_1, x_2] \\ \dots \\ f[x_{n-2}, x_{n-1}, x_n] \\ f[x_{n-1} x_n, x_n] \end{matrix}$$

Abbiamo una matrice diagonalmente dominante in senso stretto, non è simmetrica neanche nel caso dei nodi equispaziati.

Spline periodica: $s'(x_0) = s'(x_n)$ e analogamente $s''(x_0) = s''(x_n)$.

La seconda si traduce nella condizione $M_0 = M_n$. Prendendo l'espressione di s'_i, si impone l'uguaglianza e si ha $f(x_n) = f(x_0)$. Introduciamo un nodo fittizio x_{n+1} esterno all'intervallo, imponiamo che $H_n = H_0$. Imponendo tutte le condizioni si ottiene

$$\delta_n M_1 + \gamma_n M_{n-1} + 2M_n = 6f[x_{n-1}, x_n, x_{n+1}]$$

$$\gamma_1 M_n + 2M_1 + \delta_1 M_2 = 6f[x_0, x_1, x_2]$$

Quindi il sistema $A\boldsymbol{x} = \boldsymbol{b}$ è tale che:

$$A = \begin{matrix} 2 & \delta_1 & 0 & 0 & 0 & \gamma_1 \\ \gamma_2 & 2 & \delta_2 & 0 & 0 & 0 \\ 0 & \gamma_3 & 2 & \delta_3 & 0 & 0 \\ 0 & 0 & 0 & \gamma_{n-1} & 2 & \delta_{n-1} \\ \gamma_n & 0 & 0 & 0 & \delta_n & 2 \end{matrix}$$

Il vettore delle incognite è

$$\boldsymbol{x} = \begin{matrix} M_1 \\ M_2 \\ \dots \\ M_{n-1} \\ M_n \end{matrix}$$

mentre il vettore del termine noto è

$$
\begin{array}{c}
f[x_0, x_1, x_2] \\
f[x_1, x_2, x_3] \\
\dots \\
f[x_{n-2}, x_{n-1}, x_n] \\
f[x_{n-1}, x_n, x_{n+1}]
\end{array}
$$

Questa matrice $n \times n$ non è tridiagonale a causa di δ_n in posizione A_{n1} e γ_1 in posizione A_{1n}. La matrice in ogni caso è diagonalmente dominante in senso stretto. Se i nodi sono equispaziati si ha la simmetria.

5.4.4 Conclusione

Nei tre casi, la matrice è diagonalmente dominante in senso stretto, e per il primo teorema di Gersch-Gorin è non singolare.

Teorema 5.7

Se i nodi $A = x_0 < x_1 < \cdots < x_{n-1} < x_n = b$ sono distinti, allora esiste ed è unica la *spline* cubica che interpola f funzione assegnata.

Teorema 5.8

Supponiamo che f sia $C_4([a, b])$, consideriamo un partizionamento di $[a, b]$ in sottointervalli H_i, poniamo $H = \max_i H_i$ e chiamiamo $\beta = \frac{H}{\min_i H_i}$, e sia s la *spline* cubica interpolante.

$$|f_r - s_r|^\infty \leq c_r^{4-r} |f^{(4)}|$$

5.5 Approssimazione media dei minimi quadrati

5.5.1 Descrizione del problema

Supponiamo di avere le coppie di dati $(x_k, y_k)_{k=0}^m$ con nodi x_k a due a due distinti, e numerosi. Cerco

$$P_n^*(x) = \sum_{j=1}^{n} a_j^* \phi_j(x),$$

dove $\{\phi_j(x)\}_{j=0}^n$ è una base, e

$$\sum_{k=0}^{m} (y_k - p_n^*(x_k))^2 \leq \sum_{k=0}^{m} (y_k - p_n(x_k))^2$$

per ogni $p_n \in P_N$, cioè p_n^* è il polinomio che minimizza lo scarto quadratico medio. Inoltre, introducendo nella sommatoria pesi w_k positivi, in modo da dare più importanza a determinati addendi rispetto ad altri, si ha

$$\sum_{k=0}^{m} \omega_k (y_k - p_n^*(x_k))^2 \leq \sum_{k=0}^{m} (y_k - p_n(x_k))^2 \quad \text{formula 1}$$

lo scopo è quello di minimizzare lo scarto quadratico medio pesato.

5.5.2 Determinazione del sistema

Consideriamo per semplicità la base canonica dell'insieme dei polinomi, quindi $\phi_j = x^j$. Consideriamo

$$p_n(x) = \sum_{j=0}^{n} a_j \phi_j(x)$$

e inserendo l'espressione di p_n nel lato destro della formula 1:

$$\sum_{k=0}^{m} \omega_k (y_k - \sum_{j=0}^{n} a_j \phi_j(x_k))^2 = \Phi(a_0, \ldots, a_n)$$

che è un funzionale nei parametri da determinare. Cerchiamo il valore dei parametri per cui otteniamo la minimizzazione di p_n.

Il minimo rende stazionario il gradiente di Φ, quindi calcoliamo le derivate parziali. Per ogni $i = 0, \ldots, n$, si ha

$$\frac{\partial \Phi}{\partial a_i} = \sum_{k=0}^{m} \omega_k * 2 * (y_k - \sum_{j=0}^{n} a_j \phi_j(x_k)) * (-\phi_i(x_k)) = 0$$

$$\sum_{k=0}^{m} \omega_k \sum_{j=0}^{n} a_j \phi_j(x_k) * \phi_i(x_k) = \sum_{k=0}^{m} w_k y_k \phi_i(x_k)$$

e invertendo le sommatorie

$$\sum_{j=0}^{n} (\sum_{k=0}^{m} \omega_k \phi_j(x_k) * \phi_i(x_k)) a_j = \sum_{k=0}^{m} \omega_k y_k \phi_i(x_k)$$

Abbiamo un sistema lineare di $n+1$ equazioni nelle $n+1$ incognite a $a_0, \ldots, a_n$. Riscrivendo il tutto in maniera compatta abbiamo il sistema:

$$C\boldsymbol{a} = \boldsymbol{b}$$

dove

$$C = (C_{ij}) = \sum_{k=0}^{m} \omega_k \phi_j(x_k) \phi_i(x_k),$$

mentre

$$b_i = \sum_{k=0}^{m} \omega_k y_k \phi_i(x_k), \quad i, j = 0, \ldots n$$

Per dimostrare che la soluzione di tale sistema è un minimo, analizziamo la matrice $C = H_\Phi$. Dobbiamo verificare che l'hessiana è definita positiva e simmetrica.

Proposizione 5.1

La matrice $C = H_\Phi$ è definita positiva (e di conseguenza esiste unica la soluzione del problema, e a $\boldsymbol{a}$ è un minimo per Φ.)

Dimostrazione

Fattorizziamo la matrice C per decomporla in pezzi, nella forma

$$C = P^t W P$$

P^t è una matrice rettangolare di $n+1$ righe e $m+1$ colonne, con $m >> n$, W è una matrice quadrata $m+1 \times m+1$, mentre P è $(m+1) \times (n+1)$.

P è definita nel modo seguente.

$$P_{st} = \phi_t(x_s)$$

$$W = \mathrm{diag}(w_1, \ldots, w_{m+1})$$

La matrice della forma $P^t W P$ è detta matrice delle equazioni normali pesate. Il termine noto si può scrivere come $P^t W Y$.

Dimostriamo che C è simmetrica.

$$C^t = (P^t W P)^t = P^t W^t P = P^t W P$$

perché W è diagonale.

Dimostriamo che C è definita positiva.

Per ogni $\boldsymbol{x} \neq 0$, dobbiamo dimostrare che

$$\boldsymbol{x}^t C \boldsymbol{x} > 0$$

$$\boldsymbol{x}^t C \boldsymbol{x} = \boldsymbol{x}^t P^t W P \boldsymbol{x}$$

e definendo $\boldsymbol{y} = P\boldsymbol{x}$ si ha

$$\boldsymbol{x}^t C \boldsymbol{x} = \boldsymbol{y}^t W \boldsymbol{y}$$

Supponiamo che P abbia rango massimo. Allora $\boldsymbol{y} \neq 0$, e siccome W è definita positiva per definizione, $\boldsymbol{y}^t W \boldsymbol{y} > 0$.

Dimostriamo che P ha rango massimo.

Caso particolare: consideriamo $\phi_j(x) = x^j$. Allora

$$P_{s,t} = (x_k^j), k = 0, \ldots, m, j = 0, \ldots, n$$

Per esteso

$$P = \begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_m & x_m^2 & \ldots & x_m^n \end{pmatrix}$$

Questa è la sottomatrice principale della matrice di Vandermonde. Siccome i nodi sono a due a due distinti, la Vandermonde è non singolare, allora ogni sua sottomatrice ha colonne linearmente indipendenti. P ha rango massimo.

Considerando per lo spazio P_N una qualsiasi altra base invece di quella canonica, vale la medesima proprietà.

Infatti la matrice del cambiamento di base è non singolare e quindi il rango è conservato.

Nel caso della base canonica dei polinomi,

$$C = \sum_{k=0}^{m} \omega_k x_k^{i+j}$$

In particolare:

$$C_{00} = \sum_{k=0}^{m} \omega_k$$

$$C_{10} = \sum_{k=0}^{n} \omega_k x_k = C_{01}$$

$$C_{20} = \sum_{k=0}^{m} \omega_k x_k^2 = C_{11} = C_{02}$$

Abbiamo una matrice con elementi costanti lungo le antidiagonali. Una matrice con questa struttura viene chiamata matrice di Henkel.

5.5.3 Ricerca della soluzione del sistema

Data la scrittura $C = P^t W P$, l'approssimazione ai minimi quadrati, come impostato inizialmente, consiste nella risoluzione del sistema sovradeterminato $P\boldsymbol{a} = \boldsymbol{y}$ di $m+1$ equazioni in $n+1$ incognite. Infatti, per minimizzare la norma del residuo, risolviamo

$$\sum_{j=0}^{n} a_j \phi_j(x_k) = y_k, \quad k = 0, \ldots, m$$

Non c'è una soluzione classica di questo sistema, ma dobbiamo cercare una soluzione più debole.

Per minimizzare la norma euclidea del residuo, vogliamo determinare il vettore $\boldsymbol{a}^*$ tale che

$$\phi(\boldsymbol{a}^*) = (|\boldsymbol{y} - P\boldsymbol{a}|)^2 = \min_{\boldsymbol{a} \in \mathbb{R}^{n+1}} (|\boldsymbol{y} - P\boldsymbol{a}|)^2$$

Invece di determinare la soluzione di $P\boldsymbol{a} = \boldsymbol{y}$ cerchiamo la soluzione di

$$P^t P\boldsymbol{a} = P^t \boldsymbol{y} = \boldsymbol{b}$$

(sistema delle equazioni normali). Se supponiamo di avere pesi ω_k generici, possiamo definire

$$w(R) = R^t W R = \sum_k \omega_k r_k^2$$

Possiamo pensare questa norma come la norma euclidea al quadrato di $W^{1/2}R$.

5.5.4 Possibilità per la risoluzione del sistema dei minimi quadrati

Prima possibilità: consideriamo la fattorizzazione di Cholesky, per matrici simmetriche e definite positive come C.

Risolviamo quindi i sistemi

$$R^t \boldsymbol{z} = \boldsymbol{b}$$

$$R\boldsymbol{a} = \boldsymbol{z}$$

Consideriamo per semplicità $w_k = 1$.

Analizziamo il costo computazionale nella costruzione di C:

$$C_{ij} = \sum_{k=0}^{m} \omega_k \phi_j(x_k)\phi_i(x_k)$$

C ha n^2 elementi, e ognuno richiede m prodotti. C è simmetrica quindi basta calcolare metà degli elementi, per un totale di $n^2 * m/2$ operazioni.

Siccome il costo della fattorizzazione di Cholesky di C è $n^3/6$, in totale abbiamo: $n^2/2 * (m + n/3)$ operazioni. Quando i dati sono molti (m grande) si hanno problemi di accuratezza e di rappresentabilità nella fattorizzazione di Cholesky.

5.5.5 Fattorizzazione QR per la risoluzione del sistema dei minimi quadrati

Supponiamo di avere $A \in C_{m\times n}$ di rango massimo (nel problema corrisponde a P). Calcoliamo la fattorizzazione QR di A.

Considerata $Q_h = p_1 p_2 \dots p_n$ (nel caso di matrici rettangolari abbiamo n colonne da annullare), otteniamo una matrice R con m righe e n colonne, suddivisa in due blocchi: R_1 triangolare superiore e r_2 blocco trapezoidale di zeri delle sottocolonne annullate.

Il blocco superiore è sicuramente non singolare, perché A ha rango massimo.

$$|A\boldsymbol{x} - \boldsymbol{b}|_r = |QR\boldsymbol{x} - \boldsymbol{b}|$$

$$= |Q(R\boldsymbol{x} - QH\boldsymbol{b})|$$

ma siccome Q è unitaria conserva la norma 2:

$$= |R\boldsymbol{x} - Q_h\boldsymbol{b}|$$

e ponendo $Q_h\boldsymbol{b} = \boldsymbol{c}$, siccome R ha due blocchi e il secondo è nullo, e siccome $\boldsymbol{c} = (\boldsymbol{c}_1, \boldsymbol{c}_2)$, si ha

$$|R_1\boldsymbol{x} - \boldsymbol{c}_1 - \boldsymbol{c}_2|_2$$

Cerchiamo

$$\min_{x\in\mathbb{C}^n} |A\boldsymbol{x} - \boldsymbol{b}|^2$$

che per quanto appena osservato è

$$= \min |R_1\boldsymbol{x} - \boldsymbol{c}|^2$$

$$(|R_1\boldsymbol{x} - \boldsymbol{c}_1 - \boldsymbol{c}_2|)^2 \leq |R_1\boldsymbol{x} - \boldsymbol{c}_1|^2 + |\boldsymbol{c}_2|_2^2$$

$$\min_{\boldsymbol{x}\in\mathbb{C}^n} (|R_1\boldsymbol{x} - \boldsymbol{c}_1|)^2 + |\boldsymbol{c}_2|^2$$

Il primo addendo è minimo quando

$$R_1\boldsymbol{x} = \boldsymbol{c}_1$$

e rimane

$$\min |\boldsymbol{c}_2|^2$$

Si conclude che, data la matrice A, si calcola la fattorizzazione QR e si risolve quindi

$$R_1\boldsymbol{x} = \boldsymbol{c}_1$$

Il costo è $n^3/3$ per la fattorizzazione e mn^2 per la risoluzione del sistema lineare, e in totale

$$n^2 * (m - n/3)$$

che è maggiore di quello della fattorizzazione di Cholesky. Questo metodo però è più sicuro della fattorizzazione di Cholesky.

5.5.6 Relazione tra i due metodi

Tornando al problema di approssimazione, ci si chiede qual è la relazione tra R_1 che viene dall'applicazione della QR ad A generica e $\bar{R}$ che otteniamo facendo la Cholesky. Queste matrici sono uguali a meno di matrici di fase (diagonali unitarie).

$$P^t W P = \bar{R}_h \bar{R} = R^h Q_h Q R = R_1^h R_1$$

e R_1 differisce per una matrice di fase da $\bar{R}$.

Nel caso di pesi diversi da 1, si deve considerare la norma euclidea pesata e si può ripetere il procedimento analogo.

Consideriamo la norma 2 di

$$W^{1/2}(A\boldsymbol{x} - \boldsymbol{b})$$

quindi consideriamo QR di $\bar{A} = W^{1/2} * A$.

5.5.7 Condizionamento

Applicando la QR ad A, $K(R_1) = K(R) = K(A)$.

Costruendo $C = A_h A$ con la Cholesky, si ha $K(A_h A) = K(A)^2$.

Quindi le equazioni normali hanno lo svantaggio di un indice di condizionamento quadrato rispetto a quello di A.

5.6 Approssimazione dei minimi quadrati con funzioni continue

5.6.1 Descrizione del problema

Supponiamo di avere una funzione f continua su un certo intervallo $[a, b]$ (invece che un insieme di dati), e supponiamo di considerare un set di funzioni $\{\phi_j(x)\}$ che costituiscono un insieme di generatori indipendenti. Vogliamo determinare $\phi_*(x) = \sum_{j=1}^{n} \alpha_j^* \phi_j(x)$ tali che

$$(|f - \phi_*|)^2 = \min_{\phi \in V}(|f - \phi|)^2$$

Consideriamo il prodotto scalare L_2 pesato, tale che

$$g \cdot h = \int_a^b \omega(x) g(x) h(x)\, dx$$

dove $\omega(x)$ viene chiamata peso.

Si dice funzione peso una funzione $\omega(x) \geq 0 \forall x \in [a, b]$, e tale che $\int_a^b \omega(x) dx < \infty$. Il prodotto scalare induce una norma

$$|g| = \sqrt{\int_a^b \omega(x) g(x)\, dx}$$

Chiamiamo

$$\psi(a_0, \ldots, a_n) = (|f - \phi|)^2 = \int_a^b [f(x) - \sum_{j=0}^{n} a_j \phi_j(x)]^2 \omega_x\, dx$$

Per ogni $i = 0, \ldots, n$ consideriamo

$$\frac{\partial \psi}{\partial a_i} = \int_a^b \frac{\partial}{\partial a_i} [f - \sum_{j=0}^{n} \phi(x)^2] \omega(x)\, dx$$

$$= \int_a^b 2 * (f(x) - \sum_{j=0}^{n} a_j \phi_j(x)) * (-\phi_j(x)) * \omega(x)\, dx$$

e imponiamo

$$2(-f \cdot \phi_i + \sum_{j=0}^{n} a_j * \phi_i \cdot \phi_j) = 0$$

Otteniamo il sistema delle equazioni normali

$$C\boldsymbol{a} = \boldsymbol{b}$$

con

$$C = (c_{ij}) = \phi_i \cdot \phi_j$$

$$b_i = f \cdot \phi_i$$

Come nel caso discreto, $C = H_\psi$, e dimostriamo che C è simmetrica e definita positiva.

Proposizione 5.2

Se C è simmetrica definita positiva, allora esiste unica la soluzione del sistema

$$C\boldsymbol{a} = \boldsymbol{b}$$

ed è un minimo.

Dim. La simmetria è ovvia per le proprietà del prodotto scalare

$$c_{ij} = \phi_i \cdot \phi_j = \int_a^b \phi_i \phi_j \omega(x)\, dx = \phi_j \cdot \phi_i \forall i, j$$

Dimostriamo che per ogni $z \neq 0$.

$$\boldsymbol{z}^t C \boldsymbol{z} > 0$$

dove

$$\boldsymbol{z}^t C \boldsymbol{z} = \sum_{i,j=0}^{n} z_i C_{ij} z_j$$

e sostituendo le espressioni dei C_{ij} otteniamo

$$= (\sum_i z_i \phi_i) * \sum_j \phi_j z_j)$$

$$= (|\sum_{j=0}^{n} \phi_j z_j|)^2 \geq 0$$

$z^t C z = 0$ se e solo se $|\phi| = 0$, cioè

$$\sum_{j=0}^{n} \phi_j z_j = 0$$

e quindi se e solo se $z_j = 0 \forall j$.

Si garantisce che l'unica soluzione del sistema lineare è un minimo.

5.6.2 Esempio sullo spazio dei polinomi

Consideriamo come peso $w(x) = 1$, e i generatori $\phi_j = x^j, \quad j = 0, \ldots, n$ (spazio dei polinomi di grado n).

$$c_{ij} = \int_a^b x^i x^j\, dx = \int_a^b x^{i+j}\, dx = 1/(i+j+1) * [b^{i+j+1} - a^{i+j+1})$$

Si ottiene una matrice di Henkel, costante lungo le antidiagonali.

Nel caso in cui $[a, b] = [0, 1]$ si ha $c_{ij} = 1/(i+j+1)$ e C è la matrice di Hilbert.

$$\begin{matrix} 1 & 1/2 & 1/3 & 1/4 \\ 1/2 & 1/3 & 1/4 & 1/8 \\ 1/3 & 1/4 & 1/5 & 1/6 \\ 1/4 & 1/5 & 1/6 & 1/7 \end{matrix}$$

È un esempio di matrice mal condizionata.

5.6.3 Sistema di generatori ortogonali

Per evitare di avere sistemi con matrici mal condizionate, consideriamo $\phi_i \cdot \phi_j = 0 \forall i \neq j$, in modo che C sia una matrice diagonale.

Introduciamo famiglie di polinomi ortogonali rispetto a funzioni peso assegnate.

5.6.4 Caso 1: polinomi di Legendre

Questi polinomi sono ortogonali sull'intervallo $[-1, 1]$ con $\omega(x) = 1$.

$$l_i \cdot l_j = \int_a^b l_i l_j \, dx = k * \delta_{ij}$$

$$l_0(x) = 1$$

$$l_1(x) = x$$

$$(j+1) * l_{j+1}(x) = (2j+1)x * l_j(x) - j l_{j-1}(x) \quad \text{formula di ricorsione a tre termini}$$

dove $l_i \cdot l_i = 2/(2i+1)$.

Segue che, siccome il sistema dei minimi quadrati ha una matrice diagonale, si ha

$$a_j = \int_{-1}^{1} \frac{f * \phi_j}{2/(2j+1)}$$

Il condizionamento spettrale di una matrice simmetrica e definita positiva è

$$\frac{\lambda_{max}(C)}{\lambda_{min}(C)} = \frac{2/(0+1)}{2/(2n+1)} = 2n+1$$

e il condizionamento cresce in maniera lineare rispetto a n.

5.6.5 Caso 2: polinomi di Chebyshev

Definiamo i pesi

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

e

$$t_j(x) = \cos(j * \arccos x)$$

$$t_i \cdot t_j = \int_{-1}^{1} \frac{\cos(j * \arccos(x)) * \cos(i * \arccos(x))}{\sqrt{1-x^2}} =$$

$$= \begin{cases} 0 & \iff i \neq j \\ \pi & \iff i = j = 0 \\ \pi/2 & \iff i = j > 0 \end{cases}$$

$$a_j = \int_{-1}^{1} \frac{f * t_j}{\sqrt{1-x^2} * (t_j \cdot t_j)}$$

Cerchiamo

$$K_2(C) = \frac{\pi}{\pi/2} = 2$$

e il condizionamento è indipendente dalla dimensione.

6

Formule di quadratura

6.1 Formule di quadratura approssimate

6.1.1 Descrizione del problema

Data una funzione $f\colon [a,b] \to \mathbb{R}$, si vuole calcolare

$$I_f = \int_a^b f(x)\,dx$$

Siccome per ipotesi f è continua, per il teorema fondamentale del calcolo esiste una primitiva F tale che

$$\int_a^b f(x)\,dx = F(b) - F(a)$$

Approssimiamo I_f con

$$I_n(f) = \sum_{i=0}^{n} w_i * f(x_i) \quad \text{formula } *$$

dove gli x_i vengono chiamati **nodi** e i w_i vengono chiamati **pesi**. È quindi possibile calcolare gli integrali anche nel caso in cui si conoscono solo alcuni dati e non l'espressione della funzione.

◊ **Definizione** Si definisce **errore analitico** o **resto** la quantità $E_n(f) = I(f) - I_n(f)$.

◊ **Definizione** Si definisce **grado di precisione** di una formula di quadratura

$$\max\{k \in \mathbb{Z}\, t.c.\, I_n(f) = I(f),\, \forall f \in P_k\}$$

con P_k insieme dei polinomi di grado k.

Si richiede che le costanti vengano sempre integrate esattamente, cioè $I_n(1) = I(1)$, quindi imponiamo la proprietà di consistenza:

$$\sum_{i=0}^{n} w_i = \int_a^b 1 = b - a$$

Se questa proprietà non è soddisfatta la formula non è un buon algoritmo per il calcolo degli integrali.

6.1.2 Formule di quadratura con interpolazione

Si pensa di approssimare f con un polinomio di grado n, usando le tecniche di interpolazione. Nella pratica si applicano le formule di quadratura in forma composta (integrazione in sottointervalli).

Si pone

$$\begin{aligned} I_n(f) &= \int_a^b \sum_{i=0}^{n} f(x_i) * l_i(x) \\ &= \sum_{i=0}^{n} f(x_i) \int_a^b L_i(x)\, dx \end{aligned}$$

dove l'integrale è il peso w_i definito nella formula di quadratura $*$.

Esistono due possibilità:

1. x_i sono nodi equispaziati (formule di Newton-Cove),
2. x_i sono nodi corrispondenti agli zeri di polinomi ortogonali (formule gaussiane).

6.2 Formule di Newton-Cotes

6.2.1 Formula dei trapezi

Consideriamo una funzione positiva crescente. Consideriamo la retta passante per $(a, f(a))$ e $(b, f(b))$ e calcoliamo l'area del trapezio sotteso, ponendo $h = b - a$. Il trapezio ha area

$$h/2 * (f(a) + f(b)) * (b - a) = I_n(f)$$

Osserviamo che il polinomio interpolante f nei nodi $x_1 = a$ e $x_2 = b$ è:

$$f(a) * l_1(x) + f(b) * l_2(x) = f(a) * \frac{x-a}{b-a} + f(b) * \frac{x-b}{a-b}$$

quindi

$$\int_a^b f(x)\, dx$$

si approssima con

$$I_n = \int_a^b f(a) * \frac{x-a}{b-a} + f(b) * \frac{x-b}{a-b}\, dx =$$

$$I_n = \int_a^b f(a) * \frac{x-a}{b-a} - f(b) * \frac{x-b}{b-a}\, dx =$$

$$I_n = \frac{1}{b-a} * \int_a^b f(a) * (x-a) - f(b) * (x-b)\, dx =$$

$$I_n = \frac{1}{b-a} * \int_a^b x(f(a) - f(b)) - af(a) + bf(b)\, dx =$$

$$I_n = \frac{1}{b-a} * [x^2/2(f(a) - f(b)) - af(a)x + bf(b)x]_a^b =$$

$$I_n = \frac{1}{b-a} * [b^2/2f(a) - a^2/2f(a) - b^2/2f(b) + a^2/2f(b) - abf(a) + a^2f(a) + b^2f(b) - abf(b)] =$$

$$I_n = \frac{1}{b-a} * [b^2/2f(a) + a^2/2f(a) + b^2/2f(b) + a^2/2f(b) - abf(a) - abf(b)] =$$

$$I_n = \frac{1}{2(b-a)} * [b^2f(a) + a^2f(a) + b^2f(b) + a^2f(b) - 2abf(a) - 2abf(b)] =$$

$$I_n = \frac{1}{2(b-a)} * [(b-a)^2f(a) + (b-a)^2f(b)]$$

$I_n = 1/2[(b-a)f(a) + (b-a)f(b)] = (b-a)/2 * (f(b) + f(a))$ formula dei trapezi

Supponiamo che $f \in C^2([a,b])$, allora

$$f(x) - p_1(x) = \frac{f^{(2)}(\psi)}{2!} * w_2(x), \quad \psi = \psi(x)$$

(deriva dall'espressione dell'errore analitico per l'interpolazione di polinomi).

Quindi si ha

$$E_{an} = I(f) - I_n(f) = \int_a^b f(x) - p_1(x)\, dx$$

$$= 1/2 * \int_a^b f^{(2)}(\psi(x)) * W_2(x)\, dx$$

Nel caso della formula dei trapezi, $w_2(x) = (x-a) * (x-b)$ ha segno negativo sull'intervallo $[a,b]$. Possiamo quindi applicare il teorema del valor medio.

Teorema 6.1

Supponiamo di avere $G\colon [a,b] \to \mathbb{R}$ continua, $\phi\colon [a,b] \to \mathbb{R}$ integrabile, che non cambia segno in $[a,b]$. Allora esiste $\psi \in [a,b]$ tale che

$$\int_a^b g(x)\phi(x)\, dx = g(\psi) * \int_a^b \phi(x)\, dx$$

Allora, riconsiderando la formula dell'errore, siccome la funzione peso non cambia segno, si applica il teorema del valor medio. Esisterà ψ indipendente da x tale che

$$E_{an} = 1/2 * f^{(2)}(\psi) * \int_a^b w_2(x)\, dx$$

$$E_{an} = 1/2 * f^{(2)}(\psi) * \int_a^b (x-a)(x-b)\, dx$$

$$E_{an} = 1/2 * f^{(2)}(\psi) * \int_a^b x^2 - (a+b)x + ab\,dx$$

$$E_{an} = 1/2 * f^{(2)}(\psi) * [x^3/3 - (a+b)x^2/2 + abx]_a^b =$$

$$E_{an} = 1/2 * f^{(2)}(\psi) * [b^3/3 - a^3/3 + (a+b)a^2/2 - (a+b)b^2/2 + ab^2 - a^2b] =$$

$$E_{an} = 1/2 * f^{(2)}(\psi) * [b^3/3 - a^3/3 + a^3/2 + ba^2/2 - ab^2/2 - b^3/2 + ab^2 - a^2b] =$$

$$E_{an} = 1/2 * f^{(2)}(\psi) * [a^3/6 - ba^2/2 + ab^2/2 - b^3/6] =$$

$$E_{an} = \frac{1}{12} * f^{(2)}(\psi) * [a^3 - 3ba^2 + 3ab^2 - b^3] =$$

$$E_{an} = \frac{1}{12} * f^{(2)}(\psi) * (b-a)^3$$

$$E_{an} = \frac{H^3}{12} * f^{(2)}(\psi)$$

6.2.2 Formula dei trapezi composita

Se l'intervallo è grande, l'errore potrebbe essere elevato, quindi si pensa di passare alla formula composita. Suddividiamo quindi $[a, b]$ in m sottointervalli, consideriamo la partizione

$$a = x_0 < x_1 < \cdots < x_m = b$$

Poniamo $H = (b-a)/m$ (ipotesi semplificativa: intervalli di uguale ampiezza). Si calcola l'area del trapezio su ogni intervallino, e sommando i vari contributi otteniamo:

$$I_{k,m}(f) = H/2 * \sum_{k=0}^{m-1} [f(x_k) + f(x_{k+1})]$$

e siccome i nodi interni compaiono due volte possiamo riscriverlo come

$$I_{k,m}(f) = H * (f(a)/2 + \sum_{i=1}^{m-1} f(x_i) + f(b)/2)$$

H è la distanza tra due nodi successivi, e più la partizione è fine, più l'approssimazione è buona.

Sommando i vari contributi dell'errore analitico, si ha:

$$E_{1,m}(f) = -\sum_{k=0}^{m-1} \frac{f^{(2)}(\psi_k)}{12} * H^3$$

(come dimostrato prima $H^3/6$ è l'integrale del polinomio nodale) e moltiplicando e dividendo per m:

$$= -m\frac{H^3}{12} * \sum_{k=0}^{m-1} f^{(2)}(\psi(k))/m$$

e per il teorema del valor medio discreto, assumendo che $f^{(2)}(x)$ sia continuo,

$$= -H^3/12 * m * f^{(2)}(\psi),\ \psi \in [a, b]$$

$$= -H^2/12 * H * m * f^{(2)}(\psi)$$

e siccome $Hm = h$, otteniamo il risultato cercato.

$$E_{k,m}(f) = -h/12 * H^2 * f^{(2)}(\psi)$$

6.2.3 Formula di Simpson

Consideriamo la funzione $f\colon [a, b] \to \mathbb{R}$, sia x_1 il punto medio dell'intervallo. Considerando il polinomio di grado 2 interpolante nei tre nodi equispaziati, si ha

$$I_2(f) = H/3 * (f(a) + 4f((a + b)/2) + f(b))$$

con $h = (b - a)/2$ (passo di equispaziatura dei nodi).

$$I_2(f) = \int_a^b p_2(x)\, dx$$

$$= \int_a^b \sum_{i=0}^{2} f(x_i) * L_i(x)\, dx$$

$$= \sum_i f(x_i) * \int_a^b L_i(x)\, dx$$

Considerando i polinomi di Lagrange di grado 2 otteniamo i pesi $w_1 = w_3 = H/3$ e $w_2 = 4H/3$.

6.2.4 Modalità di scrittura dei pesi

Osservazione 6.1

Consideriamo un esempio di formula di quadratura con nodi equispaziati. I pesi della formula di quadratura si possono sempre scrivere come $H\alpha_i$ dove gli α_i non dipendono dall'intervallo $[a, b]$.

Possiamo scrivere i nodi come

$$x_i = a + bh,\ h = (b - a)/n$$

Il polinomio generico di grado n si scrive come

$$\sum_{i=0}^{n} y_i * L_i(x)$$

dove

$$L_i(x) = \prod_{j \neq i, j=0}^{n} \frac{x - x_j}{x_i - x_j}$$

Consideriamo un cambio di variabili.

$$x = \phi(t) = x_0 + th$$

tale che $\phi(0) = a, \phi(n) = b$.

L'espressione del polinomio di Lagrange diventa

$$\prod_{j \neq i} (x - x_j)/(x_i - x_j) = \prod_{j \neq i} \frac{x_0 + th - (x_0 + hj)}{x_0 + ih - (x_0 + jh)} = \prod_{j \neq i} (t - j)/(i - j)$$

Tenendo conto che $dx = hdt$, il peso si può quindi riscrivere come

$$w_i = \int_a^b L_i(x)\, dx = \int_0^n L_i(t) h dt$$

e definisco

$$\alpha_i = \int_0^n L_i(t)\, dt$$

in modo che $w_i = \alpha(t)h$ con $h = (b - a)/n$ nel caso di nodi equispaziati.

Inoltre $\alpha_u = \alpha_{n-u}$, cioè i pesi sono simmetrici. Questo deriva dal fatto che anche i nodi equispaziati sull'intervallo $[a, b]$ sono simmetrici.

6.2.5 Metodo dei coefficienti indeterminati

Fissiamo $x_0, x_1, \ldots, x_k$ nodi distinti, e imponiamo che la formula di quadratura abbia grado di precisione k.

Inoltre poniamo $f(x) = x^j$, allora per la formula di quadratura deve valere:

$$\sum_{i=0}^{m} w_i f(x_i) = \int_a^b f(x)\, dx$$

che si può riscrivere come

$$\sum_{i=0}^{m} w_i x_i^j = \int_a^b x^j\, dx = \frac{b^{j+1} - a^{j+1}}{j+1}, \quad j = 0, \ldots, k$$

Otteniamo il sistema lineare

$$\begin{matrix} 1 & 1 & \ldots & 1 \\ x_0 & x_1 & \ldots & x_k \\ x_0^2 & x_1^2 & \ldots & x_k^2 \\ \ldots & \ldots & \ldots & \ldots \\ x_0^k & x_1^k & \ldots & x_n^k \end{matrix} \times \begin{matrix} w_0 \\ w_1 \\ \ldots \\ w_k \end{matrix} = \boldsymbol{b}$$

dove il termine noto ha componenti

$$b_j = (b^{j+1} - a^{j+1})/(j+1)$$

La matrice a primo membro è la matrice di Vandermonde, e se i nodi sono distinti $\det A \neq 0$ esiste unica la soluzione.

Esempio 6.1 (metodo di Simpson)

Consideriamo un intervallo $[0,1]$ con nodi $x_0 = 0$, $x_1 = 1/2$, $x_2 = 1$. La distanza h fra i nodi è $1/2$.

Il sistema corrispondente è:

$$\begin{matrix} 1 & 1 & 1 \\ 0 & 1/2 & 1 \\ 0 & 1/4 & 1 \end{matrix} \times \begin{matrix} w_0 \\ w_1 \\ w_2 \end{matrix} = \begin{matrix} 1 \\ 1/2 \\ 1/3 \end{matrix}$$

Risolviamo il sistema:

$$\begin{cases} w_1 + w_2 + w_3 = 1 \\ 1/2w_2 + w_3 = 1/2 \\ 1/4w_2 + w_3 = 1/3 \end{cases}$$

$$\begin{cases} w_1 + w_2 + w_3 = 1 \\ 1/2w_2 - 1/2 = 1/4w_2 - 1/3 \end{cases}$$

$$\begin{cases} w_1 + w_2 + w_3 = 1 \\ 1/4w_2 = 1/6 \end{cases}$$

$$w_2 = 2/3,\ w_3 = -1/2 * 2/3 + 1/2 = 1/6$$

$$w_1 = 1 - w_2 - w_3 = 1 - 2/3 - 1/6 = 1/6$$

Siccome $h = 1/2$, si ha

$$\alpha_0 = \alpha_2 = 1/3,\ \alpha_1 = 4/3$$

e si ritrova la formula di Simpson.

6.2.6 Grado di precisione

Teorema 6.2

Supponiamo $f \in C^4([a,b])$, allora

$$E_2(f) = -h^5/90 * f^{(4)}(\xi), \quad h = (b-a)/2, \quad \xi \in [a,b]$$

Questa formula ha grado di precisione 3 (tutti i polinomi di grado 3 hanno derivata quarta nulla, e quindi errore nullo)

Tutte le formule interpolatorie su nodi equispaziati, quando il polinomio interpolante ha grado pari, hanno un grado di precisione in più di quanto ci si aspetta. Questo deriva dalla simmetria nei nodi.

Vale in generale che dato $P_{2,n}$ come polinomio interpolante, allora il grado di precisione k è $2n+1$.

6.2.7 Formula di Simpson composita

Definiamo $H = (b-a)/m$, e $x_k = a + kH/2$, $k = 0, \dots, 2m$. Segue che

$$w_0 = 1/6, w_1 = 4/6, w_2 = 1/6, w_3 = 4/6, w_4 = 1/6 \dots$$

$$P_{2,m} = H/6 * [f(x_0) + 2 * \sum_{s=1}^{m-1} f(x_{2s}) + 4 \sum_{s=0}^{m} f(x_{2s-1}) + f(x_m)]$$

(i punti medi vengono contati una volta, gli altri punti due volte).

$$E_{2,m}(f) = -(b-a)/180 * H^4/2 * f^{(4)}(\xi), \quad \xi \in [a,b]$$

Per $H \to 0$, l'errore decresce.

La formula di quadratura di Simpson e dei trapezi sono dette formule chiuse perché usano gli estremi dell'intervallo $[a,b]$. Altre formule vengono dette aperte perché tutti i nodi di quadratura sono interni all'intervallo.

6.2.8 Formula del punto medio

Consideriamo la funzione $f\colon [a,b] \to \mathbb{R}$, consideriamo il punto medio $(a+b)/2$ e approssimiamo la funzione con la retta per quel punto. Otteniamo quindi:

1. *formula semplice*: $I_{0,f} = H * f((a+b)/2)$, dove $H = b - a$;
2. *errore nella formula semplice*: $E_0(f) = H^3/3 f^{(2)}(\xi)$, con gradi di precisione $k = 1$;
3. *formula composita*: $I_{0,n}(f) = H * \sum_{k=0}^{n-1} f(x_k)$;
4. *errore nella formula composita*: $E_{0,m}(f) = (b-a)/24 H^2 * f^{(2)}(\xi)$, dove $H = (b-a)/m$, e $f \in C^2([a,b])$.

Consideriamo n generico.

$$I_n(f) = \int_a^b p_n(x)\, dx$$

dove p_n è il polinomio interpolante di grado n, quindi

$$I_n = \sum_i \int_a^b L_i(x)\, dx * f(x_i)$$

e $w = H\alpha(i)$.

Osservazione 6.2

Per grado del polinomio interpolante fino a 7, i coefficienti sono tutti positivi, mentre per $n > 8$ i pesi possono anche essere negativi (si può quindi avere la possibilità dell'errore di cancellazione).

Vale il seguente teorema.

Teorema 6.3

La formula di quadratura $I_n(f) = \sum_{i=0}^{n} w_i f(x_i)$ con grado di precisione almeno n, coincide con la formula di quadratura interpolatoria che usa come nodi di interpolazione $x_0, x_1, \dots, x_n$.

Dimostrazione

$$I_n(f) = \sum_{i=0}^{n} w_i f(x_i)$$

è la formula interpolatoria, con p_n polinomio interpolante di grado n, segue che il grado di precisione k è almeno n. Sia $\pi(f)$ il polinomio interpolante di grado n di f. Allora esiste unica la soluzione del sistema con matrice di Vandermonde, e quindi le due formule di quadratura coincidono.

6.2.9 Stime dell'errore a priori

◊ **Definizione** L'ordine di infinitesimo di una formula di quadratura rispetto a h è il massimo $p \in \mathbb{N}$ tale che

$$|I_n(f) - I(f)| = o(h^p)$$

Teorema 6.4

Sia $I_n(f)$ la formula di quadratura interpolatoria di Newton-Coves (nodi equispaziati e polinomio interpolante di grado n), $h = (b-a)/m$ il passo di equispaziatura. Se il grado del polinomio interpolante è pari a $f \in C_{n+2}([a,b])$ allora si dimostra che

$$E_n(f) = k_m/(n+2)! * h^{n+3} f^{(n+2)}(\xi), \xi \in [a,b]$$

(grado di precisione $n+1$ e ordine di infinitesimo $n+3$) con

$$K_m = \int_0^m w_{m+1}(t)\, dt$$

con w_{m+1} polinomio nodale, e questa quantità è negativa.

Se invece n è dispari $f \in C_{n+1}$ si ha

$$E_{n,f} = K_m/(n+1)! * H^{n+2} f^{(n+1)}(\xi)$$

(grado di precisione n e ordine di infinitesimo $n+2$) con

$$K_m = \int_0^m w_{m+1}(t)\, dt$$

Ad esempio, nel caso della formula di Simpson (tre nodi, quindi $m = 3$) dove il polinomio interpolante ha grado 2, il grado di precisione è $n+1 = 3$, e l'ordine di infinitesimo è $n+3 = 5$.

Teorema 6.5

Per le formule composite, segue che, se n è pari,

$$E_{n,m}(f) = (b-a)/(n+2)! * \frac{k_m}{n^{n+3}} H^{n+2} f^{(n+2)}(\xi)$$

e il grado di precisione è $n+1$ e l'ordine di infinitesimo è $n+2$, mentre per n dispari,

$$E_{n,m}(f) = (b-a)/(n+1)! \frac{k_m}{n^{n+2}} * H^{n+1} f^{(n+1)}(\xi)$$

e il grado di precisione è n e l'ordine di infinitesimo è $n+1$.

In ogni caso

$$E_{n,m}(f) \to 0 \iff H \to 0$$

quindi non ci sono problemi di convergenza.

6.2.10 Raddoppio del numero dei sottointervalli

Consideriamo cosa succede se si raddoppia il numero dei sottointervalli. Nel caso della formula dei trapezi:

1. Al primo passo, $m = 1$, i nodi sono a e b, e poniamo $H_1 = b - a$.
2. Per $m = 2$, consideriamo il punto medio x_1 e chiamiamo $H_2 = b - x_1$.
3. Al passo successivo, $m = 4$ (raddoppiamo il numero dei sottointervalli) e dividiamo i due sottointervalli nuovamente a metà, in modo che

$$a < x_2 < x_1 < x_3 < b$$

dove x_2, x_3 sono punti medi dei due sosttointervallini.

4. Ad ogni livello vengono introdotti nuovi nodi.

Si ha quindi

$$I_{1,1}(f) = H_1/2(f(a) + f(b))$$

$$I_{1,2}(f) = H_2/2(f(a)+2f(x_1)+f(b)) = H_2/2(f(a)+f(b)) + H_2 f(x_1) = I_{11}(f)/2 + H_2 f(x_1)$$

$$I_{1,2m}(f) = I_{1m}(f)/2 + H_{2m} * \sum_{j=1}^{m} f(x_j)$$

dove gli x_j sono i nuovi nodi.

Esercizio 6.1

Nel caso della formula di Simpson, raddoppiano i sottointervalli

$$I_{1,m}(f) = a_{2m} + b_{2m} + c_{2m}.$$

Infatti al passo 1,

$$a_1 = H_1/3(f(a) + f(b))$$

$$b_1 = 0$$

$$c_1 = 4H_1/3 f((a+b)/2) = 4H_1/3 * f(x_1)$$

Al passo 2, poniamo $H_2 = H_1/2$:

$$E_2 = H_2/3(f(a) + f(b) + 2f(x_1) + 4f(x_2) + 4f(x_3))$$

dove x_2, x_3 sono i punti medi di $[a, x_1]$ e $[x_1, b]$.

$$E_2 = H_2/3(f(a) + f(b)) + 2/3H_2 f(x_1) + 4H_2/3(f(x_2) + f(x_3)))$$

$$E_2 = H_1/6(f(a) + f(b)) + 1/3H_1 f(x_1) + 4H_2/3(f(x_2) + f(x_3)))$$

$$E_2 = a_1/2 + c_1/4 + 4H_2/3 * \sum_{i=2}^{3} f(x_i)$$

cioè

$$a_2 = a_1/2,\ b_2 = c_1/4,\ c_2 = 4H_2/3 * \sum_{i=2}^{3} f(x_i)$$

Al passo $n+1$, avendo $2m = 3 + 2^n$ nodi, si ha:

$$a_{2m} = a_m/2$$

$$b_{2m} = b_m/2 + c_m/4$$

$$c_{2m} = 4H_{2m}/3 \sum f(\text{x nuovi})$$

6.3 Integrazione automatica

6.3.1 Descrizione del problema

Problema: approssimare $I(f) = \int_a^b f(x)\,dx$ con una tolleranza assoluta ε_a o relativa ε_r assegnata.

Viene generata una sequenza di valori I_k con $k = 1, \ldots, n$, e con $\varepsilon_k = I(f) - I_k(f)$ stima dell'errore commesso. Ci si arresta al passo k tale che

$$|I(f) - I_k(f)| \leq \max\{\varepsilon_a, \varepsilon_R * |I_k(f)|\}$$

con $I_k(f)$ stima ragionevole dell'integrale.

Si possono scegliere due strategie.

1. *Strategia non adattiva*: nella procedura non adattiva $I_k(f)$ viene calcolato con una legge di distribuzione dei nodi fissata. Al passo k consideriamo m nodi equispaziati, e nel passo successivo $k+1$ ne consideriamo $2m$. Utilizziamo le formule di Newton-Coves composite.
2. *Strategia adattiva*: la distribuzione dei nodi non è fissata a priori, gli intervalli vengono partizionati a seconda delle esigenze e integrati separatamente.

6.3.2 Stima dell'errore a priori nella strategia non adattiva

Per stimare $|I(f) - I_k(f)|$, usiamo un criterio basato sull'estrapolazione di Richardson.

In base alle stime di errore a priori per le formule composite e n pari, si ha

$$I(f) - I_{n,m}(f) = (b-a)/(n+2)! * \frac{M_n}{n^{n+3}} H^{n+2} f^{(n+2)}(\xi_n)$$

Ad esempio, nel caso di Simpson $n = 2$ e si ha:

$$I(f) - I_{2,m}(f) = (b-a)/4!\frac{M_2}{2^5}H^4 f^{(4)}(\xi_m) = \delta_m(\xi_m) * H^4 \quad \text{formula 1}$$

Riscrivo la formula dell'errore con $2m$ sottointervalli:

$$I(f) - I_{2,2m}(f) = (b-a)/4!\frac{M_n}{2^5} * (H/2)^4 f^{(4)}(\xi_{2m}) = \delta_{2m}(\xi_{2m}) * \frac{H^4}{2^4} \quad \text{formula 2}$$

Sottraggo membro a membro le formule 1 e 2, e tenendo conto che

$$\delta_m(\xi_m) \simeq \delta_{2m}(\xi_{2m}) = \delta$$

segue che

$$I_{2,m} - I_{2,2m} \simeq \delta H^4 - \delta H^4/2^4 = \delta H^4 * 15/16$$

cioè

$$\delta H^4 \simeq 16/15 * |I_{2,2m}(f) - I_{2,m}(f)|$$

e sostituendo l'espressione trovata per δH^4 nella formula 2

$$I(f) - I_{2,2m}(f) \simeq 1/16 * 16/15(I_{2,2m}(f) - I_{2,m}(f)) = 1/15(I_{2,m}(f) - I_{2,2m}(f))$$

Possiamo stimare l'errore commesso con la formula di Simpson su $2m$ sottointervalli sulla base del calcolo di due formule di quadratura approssimata. Si stima che l'errore assoluto, quando si passa a $2m$ sottointervalli, si riduca di 15.

6.3.3 Stima dell'errore a posteriori nella strategia adattiva

Questa strategia è utile quando la funzione ha delle singolarità. Consideriamo un sottointervallo qualsiasi $[\alpha, \beta]$ dell'intervallo su cui si vuole integrare, e si pone

$$I(f(\alpha, \beta)) = \int_\alpha^\beta f$$

Poniamo $H_0 = (\beta - \alpha)/2$. Per il metodo di Simpson

$$s_n(f(\alpha, \beta)) = H_0/3(f(\alpha) + 4f(\alpha + H_0) + f(\beta))$$

e per l'errore si ha

$$I(f(\alpha, \beta)) - s(f(\alpha, \beta)) = -H^5/90 f^{(4)}(\xi), \xi \in [\alpha, \beta], \quad \text{formula 1}$$

Consideriamo Simpson su 2 sottointervalli, e definiamo

$$s_2 = s(f(\alpha, \alpha + \beta/2)) + s(f(\alpha + \beta/2, \beta)),$$

$$I(f(\alpha, \beta)) - s_2(f(\alpha, \beta)) = -\frac{H^5}{90} * [f^{(4)}(\xi) + f^{(4)}(\eta)], \quad \xi \in [\alpha, \alpha + \beta/2], \eta \in [\alpha + \beta/2, \beta]$$

Se assumiamo che le due derivate quarte siano circa uguali, si può scrivere

$$I(f(\alpha,\beta)) - s_2(f(\alpha,\beta)) \simeq -H^5/90 * 2 * f^{(4)}(\xi)$$

e siccome $H = H_0/2$ si ha

$$I(f(\alpha,\beta)) - s_2(f(\alpha,\beta)) \simeq -H_0^5/32 * 1/90 * 2 * f^{(4)}(\xi)$$

$$= -1/16 H_0^5/90 f^{(4)}(\xi) \quad \text{formula 2}$$

e abbiamo un fattore 16 di riduzione dell'errore.

Sottraiamo membro a membro le formule di errore 1 e 2.

$$s_2(f(\alpha,\beta)) - s(f(\alpha,\beta)) = -15/16 H_0^5/90 * f^{(4)}(\xi)$$

$$H_0^5/90 f^{(4)}(\xi) = [s(f(\alpha,\beta)) - s_2(f(\alpha,\beta))] * 16/15$$

e sostituendo nella formula 2:

$$I(f(\alpha,\beta)) - s_2(f(\alpha,\beta)) = -1/16 * 16/15 * (s(f(\alpha,\beta)) - s_2(f(\alpha,\beta)))$$

L'errore si stima come

$$\frac{-s(f(\alpha,\beta)) - s_2(f(\alpha,\beta))}{15}$$

Questa è una stima a posteriori dell'errore applicando Simpson quando lo applichiamo al generico intervallo (α,β).

6.3.4 Test d'arresto

Nella pratica, si considera il seguente test

$$|I(f(\alpha,\beta)) - s_2(f(\alpha,\beta))| \leq \frac{|s_2(f(\alpha,\beta)) - s(f(\alpha,\beta)|}{10}$$

dove chiamiamo $\varepsilon(f(\alpha,\beta))$ la quantità al numeratore.

Per ottenere la stima d'errore sull'intero intervallo $[a,b]$, ci si arresta quando

$$|\varepsilon(f(\alpha,\beta))|/10 \leq \varepsilon * (\beta - \alpha)/(b - a)$$

Infatti, se questo avviene, per l'errore totale si ha:

$$|I(f(\alpha,\beta)) - \sum_{(\alpha_i,\beta_i)\, t.c. \bigcup_i (\alpha_i,\beta_i)=(a,b)} s_2(f(\alpha_i,\beta_i))$$

$$\simeq \sum_{(\alpha_i,\beta_i)} |E_f(\alpha_i,\beta_i)|/10 \leq \sum_{(\alpha_i,\beta_i)} \varepsilon/(b-a) * |\beta_i - \alpha_i| \leq \varepsilon$$

perché la somma delle lunghezze degli intervallini è la lunghezza di (a,b).

6.3.5 Algoritmo dell'integrazione automatica adattiva

Chiamiamo A l'intervallo di integrazione attivo in cui si deve calcolare l'integrale. Chiamiamo S l'intervallo di integrazione già esaminato, in cui il test

$$|E(f(\alpha,\beta))/10| \leq \varepsilon(\beta-\alpha)/(b-a)$$

è verificato.

Chiamiamo N l'intervallo di integrazione non ancora esaminato.

Fissiamo i tre intervalli:

$$N = \emptyset$$

$$A = [a,b]$$

$$S = \emptyset$$

Al generico istante dell'algoritmo chiamiamo

$$J_s = \int_s f \simeq \int_a^\alpha f$$

$$J_A = \int_\alpha^\beta f$$

Supponiamo di aver calcolato $J(\alpha,\beta)(f)$. Se il test d'arresto è verificato, aggiungiamo a $J_s(f)$ il nuovo pezzo di integrale, altrimenti dividiamo $J_a(f)$ in due parti.

$$\text{if}(\frac{|E(f(\alpha,\beta))}{10}| \leq \varepsilon\frac{\beta-\alpha}{b-a})$$

$$S = S \cup A$$

$$A = N$$

$$\alpha = \beta$$

$$\beta = b$$

$$N = \emptyset$$

else

$$\alpha' = \alpha + \beta/2$$

$$A = (\alpha, \alpha')$$

$$\alpha = \alpha$$

$$\beta = \alpha'$$

$$N = N \cup (\alpha', \beta)$$

end

Sul calcolatore si controlla anche che (α, α') non diventi eccessivamente piccolo.

6.3.6 Function quad

La *function quad* è il metodo utilizzato da Matlab per calcolare gli integrali ed è ricorsiva.

Consideriamo il punto medio di $[a, b]$, poniamo $h = (b - a)/C$, e consideriamo i sei sottointervalli ottenuti ponendo:

$$x_1 = a,\ x_2 = a + h, x_3 = a + 2h,\ x_4 = (a + b)/2,\ x_5 = b - 2h,\ x_6 = b - h, x_7 = b$$

La distribuzione non è esattamente uniforme.

Si applica Simpson sui punti a tre a tre, e si ottiene l'approssimazione dell'integrale:

$$q = \bar{q}_1 + \bar{q}_2 + \bar{q}_3$$

dove $\bar{q}_1, \bar{q}_2, \bar{q}_3$ si ottengono applicando Simpson rispettivamente a x_1, x_2, x_3, a x_3, x_4, x_5 e a x_5, x_6, x_7.

Poniamo $\bar{q} = \int_{\bar{a}}^{\bar{b}} f$. Dividiamo $(\bar{a}, \bar{b})$ esattamente a metà, considerandone il punto medio c. Poniamo $d = \bar{a} + c/2$ $e = c/2 + \bar{b}$, $h = \bar{b} - \bar{a}$.

Per poter applicare il test d'arresto, calcoliamo

$$\bar{q}(s_1) = h/6[f(\bar{a}) + 4f(c) + f(\bar{b})]$$

e raddoppiando il numero dei nodi:

$$\bar{q}(s_2) = H/12[f(\bar{a}) + 4f(d) + 2f(c) + 4f(e) + f(\bar{b})]$$

e poniamo poi $\bar{q} - \bar{q}(s_2) = (\bar{q}(s_2) - \bar{q}(s_1))/15$ (è la stima dell'errore tra il passo 0 e il passo $2s$).

Se $|\bar{q} - \bar{q}(s_2)| < \varepsilon$, il metodo si arresta, altrimenti $(\bar{a}, \bar{b})$ viene spezzato in due parti.

$$\bar{q} = \int_{\bar{a}}^{c} f + \int_{c}^{\bar{b}} f$$

e si applica ricorsivamente la *function quad* ai due sottointervalli ottenuti.

Abbiamo una chiamata ricorsiva alla funzione, l'integrale complessivo viene calcolato come somma di due integrali.

Data una funzione con una singolarità, si spezza l'integrale in quel punto e si fa la somma degli integrali separati.

7

Problemi di Cauchy

7.1 Cenni teorici

7.1.1 Descrizione del problema

Si vuole trovare una soluzione del problema di Cauchy

$$\begin{cases} y'(t) = f(t, y) \\ y(t_0) = y_0 \end{cases}$$

con $f\colon I_0 \times S \subset \mathbb{R} \times \mathbb{R}^n$, $t_0 \in I_0, y_0 \in S$.

Teorema 7.1

Supponiamo che sia $t_0 \in J_0 \subset I_0$, dove $J_0 = U(t_0)$, sia $y_0 \in \S \subset S$, se $f \in C^0(J_0 \times \S)$, ed esiste $L \in \mathbb{R}^+$ tale che

$$|f(t_1, y_1) - f(t_2, y_2)| \leq L|y_1 - y_2|$$

per ogni $(t_i, y_i) \in J_0 \times \S$. Allora esiste unica la soluzione $Y\colon J_0 \to \S$ e si ha dipendenza continua dai dati, cioè se chiamiamo $y_s(t)$ la soluzione del problema

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(t_0) = s \end{cases}$$

segue che

$$|y_{s_1}(t) - y_{s_2}(t)| \leq e^{L(t-t_0)}|s_1 - s_2|$$

Inoltre si ha esistenza e unicità in grande della soluzione se $J_0 = I_0$ e $\S = S$.

7.1.2 Riformulazione di problemi di ordine n

Un problema di Cauchy di ordine n può essere ricondotto a un problema di Cauchy del primo ordine nel seguente modo. Dato un problema del tipo

$$\begin{cases} y^{(n)}(t) = f(t, y, y', \ldots, y^{(n-1)}) \\ f^{(i)}(t_0) = y_0^i, \quad i = 0, \ldots, n-1 \end{cases}$$

introduciamo funzioni ausiliarie:

$$z_1(t) = y(t),\ z_2(t) = y'(t),\ \ldots, z_n(t) = y^{(n-1)}(t)$$

Possiamo ricondurci a risolvere il seguente sistema con un problema di Cauchy del primo ordine:

$$\begin{cases} z_1'(t) = z_2 \\ z_2'(t) = z_3 \\ \ldots \\ z_{n-1}'(t) = z_n \\ z_n'(t) = y^{(n)}(t) = f(t, z_1, z_2, \ldots, z_n) \end{cases}$$

Se f è continua, il problema di Cauchy è equivalente all'equazione integrale

$$y(t) = y(t_0) + \int_{t_0}^{t} f(\tau, y(\tau))\, d\tau$$

7.2 Algoritmi per la risoluzione

7.2.1 Metodi a un passo

Dato il problema

$$\begin{cases} y'(t) = f(t, y) \\ y(t_0) = y_0 \end{cases}$$

supponiamo di volerlo risolvere in $I = (t_0, T) \subset I_0$.

Creiamo una successione di valori $\{y_n\}$ dove y_n approssima $y(t_n)$ e $t_n = t_0 + nh, \quad n = (T - t_0)/h$, e h viene chiamato passo di discretizzazione.

Per calcolare $y(t + h)$ conoscendo $y(t)$ consideriamo il rapporto incrementale

$$\frac{y(t + h) - y(t)}{h}$$

che deve essere approssimato. Per l'approssimazione si utilizza la derivata prima, che in questo caso è un dato del problema. Quindi si pone

$$\frac{y(t + h) - y(t)}{h} = f(t, y(t)) = y'(t)$$

ed esplicitando $y(t + h)$:

$$y(t + h) = y(t) + h * f(t, y(t)).$$

7.2.2 Metodo di Eulero esplicito

Otteniamo quindi il seguente metodo detto metodo di Eulero esplicito:

$$y_{n+1} = y_n + hf(t_n, y_n)$$

con $t_n = t_0 + nh$, e y_n approssimazione della soluzione esatta all'istante t_n.

$$h = t_{n+1} - t_n$$

Questo metodo è detto esplicito perché la quantità da calcolare, y_{n+1}, non compare nel membro di destra.

Sappiamo che

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(\tau, y(\tau))\, d\tau$$

e sostituendo alla funzione il polinomio interpolante di grado 0 in $(t_n, f(t_n, y_n))$, cioè la retta $p = f(t_n, y(t_n))$ otteniamo il metodo di Eulero esplicito:

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(t_n, y(t_n))\, d\tau$$

$$y(t_{n+1}) = y(t_n) + f(t_n, y(t_n))(t_{n+1} - t_n)$$

$$y(t_{n+1}) = y(t_n) + hf(t_n, y(t_n))$$

7.2.3 Generalizzazione del metodo

Consideriamo la generalizzazione del metodo

$$y_{n+1} = y_n + h\Phi(t_n, y_n, f(t_n, y_n); h)$$

e lo confronto con la corrispondente espressione nel continuo (soluzione esatta):

$$y(t_{n+1}) = y(t_n) + h * \Phi(t_n, y(t_n), f(t_n, y(t_n))); h)$$

All'istante t_{n+1} definisco il residuo ε_{n+1}. Ci si chiede di quanto la soluzione esatta non soddisfa lo schema numerico considerato.

$\varepsilon_{n+1} = h\tau_{n+1}$ è funzione del passo di discretizzazione, e si può scrivere

$$\tau_{n+1}(h) = \varepsilon_{n+1}/h$$

dove τ_{n+1} è detto errore di troncamento locale. ε_{n+1} è la differenza tra la soluzione $y(t_{n+1})$ e la soluzione che si ottiene considerando come y_n il valore esatto $y(t_n)$.

Più precisamente

◊ **Definizione** Si definisce **errore di troncamento locale** (LTE) la quantità

$$\tau(t, y, h) = \frac{y(t+h) - y(t)}{h} - \Phi(t, y, f, h)$$

Per dire che l'errore commesso tende a 0, non basta richiedere che $\varepsilon_{n+1} \to 0$, altrimenti si elimina la dipendenza del metodo da Φ.

7.2.4 Consistenza

◊ **Definizione** Il **metodo** è **consistente** con il problema ai valori iniziali se vale che

$$\lim_{h \to 0} \tau(t, y, h) = 0$$

con $t = t_0 + nh$. Se per ogni istante $t \in I_0$ la soluzione $y(t)$ è tale che $\tau(t, y, h) = o(h^p)$, allora il metodo è consistente di ordine p.

La condizione di consistenza è la minima condizione che dev'essere soddisfatta affinché il metodo funzioni.

Considerando il metodo di Eulero esplicito:

$$\Phi(t, y, f, h) = f(t, y(t))$$

$$\tau(t, y(t), h) = 1/h * [y(t + h) - y(t) - h\Phi(t, y, h)]$$

$$= 1/h * [y(t + h) - y(t) - hf(t, y(t))]$$

e sviluppando con Taylor $y(t + h)$ e tenendo conto che $f(t, y(t) = y'(t)$:

$$= 1/h * [y(t) + hy'(t) + h^2/2y^{(2)}(\xi) - y(t) - hy'(t)]$$

e in seguito a cancellazioni:

$$= 1/h * h^2/2y^{(2)}(\xi) = o(h)$$

e il metodo di Eulero esplicito è consistente di ordine $p = 1$.

Si ricava anche

$$\tau(h, y, f) = h/2y^{(2)}(\xi) \quad \text{formula 1}$$

7.2.5 Convergenza nell'intervallo $[t_0 T]$

Per $h \to 0$, l'insieme di valori $t_n = t_0 + nh$ tende a ricoprire l'intervallo in questione.

Chiediamo che y_n converga alla soluzione esatta $y(t)$ per ogni $t \in I$.

◊ **Definizione** Un **metodo** si dice **convergente** se per ogni problema ai valori iniziali tale che valga il teorema di esistenza e unicità, è verificata la seguente proprietà: per ogni n,

$$|y_n - y(t_n)| \leq C(h)$$

dove

$$\lim_{h \to 0} C(h) = 0$$

Il problema è convergente di ordine p se

$$C = o(h^p)$$

Disegniamo un grafico con il tempo in ascissa e la condizione y_0 in ordinata. Tenendo conto che $t = t_0 + nh$, segue che al diminuire del passo h, sono necessari più passi per raggiungere un fissato istante di tempo t. Non si può scegliere come condizione di convergenza

$$\lim_{h_i \to 0} y_n(h_i) = y_0$$

perché questo tiene conto solo della condizione iniziale e non dà importanza al fatto che dopo un fissato numero di passi, a differenti scelte di h corrispondono istanti t diversi. Bisogna quindi fissare t, e fare contemporaneamente

$$\lim_{h_i \to 0, n \to \infty} y_n(h_i) = y_0$$

Abbiamo un metodo convergente se per ogni problema ai valori iniziali

$$\lim_{h \to 0, t = t_0 + nh} y(t) = y_0$$

per ogni successione $\{y_n\}$ generata a partire da y_0.

7.2.6 Analisi di convergenza

Dato il metodo

$$y_{n+1} = y_n + hf(t_n, y_n)$$

Consideriamo l'errore all'istante t_{n+1}, e definiamo

$$e_{n+1} = y(t_{n+1}) - y_{n+1} = y(t_{n+1}) - y^*_{n+1} + y^*_{n+1} - y_{n+1}$$

dove y^*_{n+1} è il valore che otteniamo se all'istante t_{n+1} prendiamo come dato y_n il valore esatto $y(t_n)$, cioè

$$y^*_{n+1} = y(t_n) + hf(t_n, y(t_n))$$

e rappresenta l'errore locale, di consistenza.

Possiamo sostituire i primi due addendi con l'errore di troncamento moltiplicato per h, quindi

$$E_{n+1} = h\tau_{n+1} + y(t_n) + hf(t_n, y(t_n)) - y_{n+1}$$

e sostituendo l'espressione di $y_{n+1} = y_n + hf(t_n, y_n)$:

$$E_{n+1} = h\tau_{n+1} + y(t_n) + hf(t_n, y(t_n)) - y_n - hf(t_n, y_n)$$

e siccome $E_n = y(t_n) - y_n$,

$$E_{n+1} \leq h\tau_{n+1} + E_n + h * f(t_n, y(t_n)) - h * f(t_n, y_n)$$

Passando ai moduli

$$|E_{n+1}| \leq h * |\tau_{n+1}(h)| + |E_n| + h * (|f(t_n, y(t_n))| - |f(t_n, y_n)|)$$

Poniamo $\tau_h = \max_n\{\tau_{n+1}(h)\}$ e teniamo conto della lipschitzianità di f, allora otteniamo

$$\leq h|\tau_h| + |E_n| + Lh(y(t_n) - y_n) \leq h\tau_h + |E_n| + Lh * E_n \leq h\tau_h + (1 + L_h)|E_n|$$

e avendo trovato la relazione ricorsiva si prosegue fino al passo 0:

$$\leq h\tau_h + (1 + Lh) * (h\tau_h + (1 + Lh)|E_{n-1}|)$$

e siccome $E_0 \leq h\tau_h$ in conclusione si ha

$$|E_{n+1} \leq h\tau_h * (1 + (1 + Lh) + \cdots + (1 + Lh)^n)$$

$$\leq h\tau_h \sum_{k=0}^{n} (1 + LH)^k$$

$$= \frac{(1 + Lh)^{n+1} - 1}{Lh} h\tau_h$$

$$= \frac{(1 + Lh)^{n+1} - 1}{L} \tau_h$$

Siccome in generale $1 + x \leq e^x$, si ha

$$\leq \frac{(e^{hL})^{n+1} - 1}{L} * \tau_h$$

Siccome $t_{n+1} - t_0 = h$ si ha

$$= \frac{e^{L(t_{n+1} - t_0)} - 1}{L} * \tau_h$$

Se $Y \in C_2$, nel caso di Eulero esplicito si ha per la formula 1

$$\tau = h/2 f^{(2)}(\xi)$$

quindi $\tau_h \leq Mh/2$ dove $M = \max |f^{(2)}(\xi)|$.

Allora segue che

$$E_{n+1} \leq \frac{e^{L(t_{n+1} - t_0)} - 1}{L} * M/2h$$

per ogni $n \geq 0$.

La quantità tende a 0 per $h \to 0$, con lo stesso ordine 1 dell'errore di troncamento locale.

7.2.7 Metodo di Eulero implicito

Il metodo di Eulero implicito è dato da

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1})$$

Consideriamo l'errore di troncamento

$$\tau_h(y, t, n) = 1/h[y(t + h) - y(t) - h * f(t + h, y(t + h))]$$

e tenendo conto della definizione del problema di Cauchy:

$$= 1/h[y(t+h) - y(t) - hy'(t+h)]$$

$$= 1/h[y(t) + hy'(t) + h^2/2y^{(2)}(\xi) - y(t) - h(y'(t) + h * y^{(2)}(\eta))]$$

$$= 1/h * h^2(c_1 y''(\xi) + c_2 y''(\eta)$$

e abbiamo un ordine di consistenza $h = 1$.

7.2.8 Metodo dei trapezi

$$y_{n+1} = y_n + h/2(f(t_n, y_n) + f(t_{n+1}, y_{n+1}))$$

A destra ci sono due valutazioni di f e l'unica incognita è y_{n+1}.

Esercizio 7.1

Verificare, usando Taylor, che l'ordine di consistenza di questo metodo è 2.

Dim. L'errore di troncamento si definisce come

$$\tau = 1/h * [y(t+h) - y(t) - h/2(f(t_n, y_n) + f(t_{n+1}, y_{n+1})]$$

$$\tau = 1/h * [y(t+h) - y(t) - h/2y'(t) - h/2y'(t+h)]$$

$$\tau = 1/h*[y(t)+hy'(t)+h^2/2y''(t)+h^3/3y^{(3)}(\xi)-y(t)-h/2y'(t)-h/2y'(t)-h/2y''(t)h-h/2y^{(3)}(\eta)h^2/2]$$

$$\tau = 1/h * [h^3(ay^{(3)}(\xi) - by^{(3)}(\eta))] = o(h^2)$$

Considerando il metodo di Eulero implicito:

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1})$$

si ha un'equazione non lineare e può essere trattata in due modi.

1. *Definiamo il metodo di punto fisso*:

$$y_{k+1} = y_k + hf(t_0 + nh, y_k),$$

Affinché il metodo converga si richiede che $|g'(z)| < 1$. In questo caso, con $z = y_{k+1}$ si ha:

$$g'(z) = h\frac{\partial f}{\partial z} < HL$$

quindi la costante di Lipschitz influenza la scelta del passo di discretizzazione. Se la costante di Lipschitz è grande, bisogna scegliere un passo di discretizzazione piccolo per poter avere convergenza.

2. *Applichiamo il metodo di Newton.* Poniamo $z = y_{n+1}$ e cerchiamo lo zero di

$$\phi(z) = z - y_n + hf(t_{n+1}, z)$$

con il metodo

$$z_{k+1} = z_k - \frac{\phi(z_k)}{\phi'(z_k)}$$

con vettore d'innesco $z_0 = y_n$.

$$\phi'(z) = 1 - h\frac{\partial f}{\partial z}(t_{n+1})$$

Come test d'arresto si ha il test dell'incremento.

7.3 Metodi multipasso

7.3.1 Aumento dell'ordine di convergenza

Per aumentare l'ordine di convergenza del metodo ci sono due strade possibili:

1. *metodi multipasso*: usano informazioni precedenti ma rimangono lineari, ne sono esempi i metodi di Adam-Bashword (espliciti), di Adam-Multonn (impliciti), di Mistrom, i *generalized Simpson* (espliciti e impliciti), BDF usate nel caso dei problemi *stiff*;
2. *metodi a un passo non lineari.*

7.3.2 Metodi multipasso lineare

I metodi di questo tipo si definiscono come

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} b_j f_{n+j}$$

dove

$$f_{n+j} = f(t_{n+j}, y_{n+j})$$

α_j, β_j sono tali che $\alpha_k = 1$ e

$$\sum |\alpha_0| + |\beta_0| \neq 0$$

Osserviamo che se $\beta_k = 0$ il metodo è esplicito perché a secondo membro non compare la soluzione da calcolare al passo corrente, mentre diversamente è implicito, e si può pensare di applicare il metodo del punto fisso

$$y_{n+k} = h\beta_k f(t_{n+k}, y_{n+k}) + \phi$$

Dev'essere $h < 1/(|\beta_k| * L)$.

Teorema 7.2

Consideriamo i seguenti polinomi

$$\rho(z) = \sum_{j=0}^{k} \alpha_j z^j$$

con z variabile complessa (primo polinomio caratteristico).

$$\sigma(z) = \sum_{j=0}^{k} \beta_j z^j$$

e consideriamo il metodo

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h\phi_f(y_{n+k}, y_{n+k-1}, y_n, t_n, h) = h \sum_{j=0}^{k} \beta_j f_{n+j}$$

Il metodo è consistente (cioè l'errore di troncamento locale tende a 0 per $h \to 0$), se valgono le seguenti condizioni:

$$\sum_{j=0}^{k} \alpha_j = 0, \quad \longrightarrow \quad \rho(1) = 0$$

$$\sum_{j=0}^{k} j\alpha_j f(t_n, y(t_n)) = \phi_f(t_n, t_n, t_n$$

(h tende a 0 quindi tutti gli istanti tendono t_n)
con $\phi_f = \sum_j \beta_j f_{n+j}$.

Dimostrazione

Affermiamo che

$$y(t_{n+j}) - y(t_n) = jhy'(\xi_j) = jhf(\xi_j)$$

con $\xi_j \in (t_n, t_{n+j})$ con $j = 0, \dots, k$ per il teorema del valor medio.

Consideriamo l'errore di troncamento:

$$\tau = 1/h[\sum_{j=0}^{k} \alpha_j y(t_{n+j}) - h\phi_f]$$

e in base a quanto appena ricavato per y_{n+j}:

$$= 1/h[\sum_{j=0}^{k} \alpha_j * (y(t_n) + jhy'(\xi_j))] - \phi_f$$

e spezzando le somme

$$= 1/h * \sum_{j=0}^{k} \alpha_j * y(t_n) + \sum_{j=0}^{k} j\alpha_j y'(\xi_j) - \phi_f$$

Per $h \to 0$ $y'(\xi_j) \to y'(t_n)$ e $\phi_f(y'(t), y''(t), \dots) = \phi_f(t_n, t_n, t_n)$

Segue quindi che si ha consistenza se

$$\sum_{j=1}^{k} \alpha_j = 0$$

e se

$$\sum_{j=0}^{k} j\alpha_j y'(t_n) - \phi_f = 0$$

Nella pratica tenendo conto che $y'(t_n) = f(t_n, y_n)$ si impongono le condizioni di consistenza:

$$\sum_j \alpha_j = 0$$

$$\sum_j [j\alpha_j - \beta_j] = 0$$

$$\sum_j [j^2/2\alpha_j - j\beta_j] = 0$$

$$\sum_j [j^3/6\alpha_j - j^2/2\beta_j] = 0 \ldots$$

7.3.3 Relazione tra convergenza e consistenza

Se ho convergenza, allora ho consistenza, ma non vale viceversa. Per avere convergenza, oltre alla consistenza dobbiamo richiedere la 0-stabilità.

◊ **Definizione** Consideriamo un problema di Cauchy perturbato

$$\begin{cases} z' = f(t, z + \delta(t)) \\ z(t_0) = z_0 + \delta \end{cases}$$

Si dice che il problema è ben posto o totalmente stabile se per qualunque coppia di perturbazioni $(\delta(t), \delta)$ e $(\delta^*(t), \delta^*)$ tali che

$$|\delta(t) - \delta^*(t)| < |\delta - \delta^*|$$

segue che, posto $z(t), z^*(t)$ le soluzioni dei corrispondenti problemi di Cauchy, esiste una costante s positiva tale che per ogni $t \in I_0$,

$$|z(t) - z^*(t)| \leq s\varepsilon$$

Definizione corrispondente nel discreto: consideriamo un sistema di differenze perturbato.

$$\begin{cases} z_{n+1} = z_n + h\phi_f(t_n, y_n, h) + \delta_{n+1}) \\ z_0 = y_0 + \delta_0 \end{cases}$$

Il metodo è 0-stabile se, date le successioni delle perturbazioni $\{\delta_n\}_{n=0}^N$, e $\{\delta_n^*\}_{n=0}^N$ che corrispondono a soluzioni z_n per $n = 0, \ldots, N$ e z_n^* per $n = 0, \ldots, N$, se esiste una costante s positiva ed esiste $h_0 > 0$ tale che scelto $0 < H < H_0$, segue che la norma della differenza tra le due soluzioni discrete è minore di $s\varepsilon$, dove supponiamo che $|\delta_n - \delta_n^*| < \varepsilon$ (la differenza rispetto alla definizione precedente è che deve esistere H_0 tale per cui si abbia la conservazione dell'entità della perturbazione).

7.3.4 Stabilità come condizione necessaria

Questo esempio mostra che è necessario richiedere la stabilità.

Esempio 7.1

Consideriamo il problema di Cauchy

$$\begin{cases} y' = 0 \\ y(t_0) = 0 \end{cases}$$

Applicando il metodo multipasso:

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = 0$$

perché f (e quindi il secondo membro) è identicamente nullo.

$(y_1, \dots, y_{k-1})$ sono molto piccoli ma non esattamente nulli.

Questa è un'equazione alle differenze omogenea. Il polinomio

$$\rho(z) = \sum_{j=0}^{k} \alpha_j z^j$$

ha radici $z_1, \dots, z_s$ distinte rispettivamente con molteplicità $\mu_1, \dots, \mu_s$, allora la soluzione dell'equazione alle differenze è

$$y_n = \sum_{i=0}^{s} \sum_{j=0}^{\mu_{i-1}} c_{ij} z_i^n * n^j$$

(simile alla scrittura delle soluzioni nelle equazioni differenziali nel continuo).

Se $|z_j| > 1$, il metodo diverge, e lo stesso avviene nel caso in cui il modulo di z è uguale a 1 e ha molteplicità maggiore di 1.

7.3.5 *Root condition*

◊ **Definizione** Consideriamo $\rho(z)$ e chiamiamo z_i le sue radici, per $i = 1, \dots, k$. $z_1 = 1$ è detta **radice principale**, e le altre radici z_i per $i = 2, \dots, k$ sono le **radici spurie**.

◊ **Definizione** Il metodo numerico

$$\sum_j \alpha_j y_{n+j} = h \sum_j \beta_j f_{n+j}$$

soddisfa la *root condition* se tute le radici z_j del polinomio caratteristico sono in modulo ≤ 1 e quelle per cui vale che $|z_p| = 1$ hanno molteplicità algebrica 1.

Teorema 7.3

Il metodo numerico

$$\sum_{j=0}^{s} \alpha_j y_{n+j} = h \sum_j \beta_j f_{n+j}$$

è 0-stabile se e solo se vale la *root condition*.

Teorema 7.4

Il metodo

$$\sum_{j=0}^{s} \alpha_j y_{n+j} = h \sum_j \beta_j f_{n+j}$$

è convergente se e solo se è consistente e 0-stabile.

Teorema 7.5 (prima barriera di Dahlquist)

Nessun metodo lineare multipasso a k passi che sia 0-stabile può avere ordine di convergenza maggiore di $k+1$ se k è dispari, e di $k+2$ se k è pari.

7.3.6 Espressione dei polinomi caratteristici nei metodi multipasso

Nei metodi di Adam-Bashword e Adam-Multonn il polinomio caratteristico è della forma

$$\rho(z) = z^{k-1} * (z-1)$$

e $z = 1$ è la radice principale necessaria per la consistenza, e c'è poi la radice 0 con molteplicità $k-1$.

Nel Simpson generalizzato

$$\rho(z) = z^k - z^{k-2} = z^{k-2} * (z^2 - 1)$$

dove le radici sono $z_1 = 1, z_2 = -1$ con molteplicità 1, e $z = 0$ con molteplicità $k-2$.

Le BDF hanno il secondo polinomio caratteristico definito come

$$\sigma(z) = \beta^k * z^k$$

7.3.7 Metodi di Adam-Bashword

I metodi di Adam-Bashword derivano da formule di quadratura. Consideriamo l'equazione integrale

$$y(t_{n+1}) - y(t_n) = \int_{t_n}^{t_{n+1}} f(t, y(t))\, dt$$

Consideriamo il polinomio interpolante f con nodi equispaziati: in particolare, per i metodi espliciti consideriamo il polinomio interpolante di grado p nei nodi $(t_n, \ldots, t_{n-p})$, per gli Adam-Multonn consideriamo il polinomio di grado $p+1$ nei nodi $t_{n+1}, \ldots t_{n-p}$ e otteniamo la formula

$$y_{n+1} = y_n + h \sum_{j=-1}^{p} \beta_j f_{n-j}$$

Se $\beta_{-1} = 0$ il metodo è esplicito, altrimenti è implicito.

Metodi di Adam-Bashword: $y_{n+1} = y_n + hf_n$ 1 nodo, Eulero esplicito

$y_{n+1} = y_n + h/2(3f_n - f_{n-1})$ 2 nodi, trapezi esplicito

$$y_{n+1} = y_n + h/12(23f_n - 16f_{n-1} + 5f_{n-2}) \quad \text{3 nodi}$$

Adam-Multon

$$y_{n+1} = y_n + hf_{n+1} \quad \text{1 nodo, Eulero implicito}$$

$$y_{n+1} = y_n + h/2(f_{n+1} + f_n) \quad \text{2 nodi, trapezi implicito}$$

$$y_{n+1} = y_n + h/12(5f_{n+1} + 8f_n - f_{n-1}) \quad \text{3 nodi}$$

In conclusione vale il seguente teorema.

Teorema 7.6

I metodi espliciti di Adam-Bashword hanno ordine q mentre quelli di Adam-Multonn hanno ordine $p = q + 1$ a eccezione dell'Eulero per cui $q = 1$ e $p = 1./$

7.3.8 Tecnica del *predictor-corrector*

Consideriamo un metodo multipasso implicito, da risolvere con un metodo di punto fisso. Poniamo

$$z = y_{n+1}$$

e consideriamo la relazione

$$z = y_n + h\beta_{-1}f(t_{n+1}, z) + h\sum_{j=0}^{p} \beta_j f_{n-j}$$

dove il secondo addendo è noto e ne conosciamo tutte le quantità. $z_{k+1} = g(z_k)$ con $g = y_n + h\beta_{-1}f(t_{n+1}, z) + h\sum_{j=0}^{p} \beta_j f_{n-j}$, e si dà come innesco y_n soluzione all'istante precedente. Si impone $g' < 1$ quindi

$$|g'(z)| = h\beta_{-1} * |\frac{\partial f}{\partial z}| \leq h\beta_{-1}L$$

e supponendo di avere le condizioni del teorema di esistenza e unicità

$$h\beta_{-1}L < 1$$

cioè sul passo h otteniamo la condizione $h < |\beta_{-1}| * L$ e la costante di Lipschitz condiziona la convergenza del metodo di punto fisso.

Il metodo è costoso, perché a ogni iterazione del metodo di punto fisso si richiede una valutazione della funzione f. Più l'innesco fornito è buono, più velocemente viene soddisfatto il criterio d'arresto, e meno oneroso è il metodo dal punto di vista computazionale.

7.3.9 Metodo *predictor-corrector*

Il metodo *predictor-corrector* consiste dei seguenti passi:

1. *predictor* ($\mathcal{P}$): $y_{n+1}(x_0)$ è ottenuto applicando uno schema multipasso esplicito, cioè

$$y_{n+1}(0) = y_n + h \sum_{j=0}^{\bar{p}} \beta_j f_{n-j}$$

2. *evaluation* ($\mathcal{E}$):

$$f_{n+1} = f(t_{n+1}, y_{n1+}(0))$$

(ora un'espressione approssimativa di f_{n+1} è nota)

3. *corrector* ($\mathcal{C}$):

$$y_{n+1} = y_n + h\beta_{-1} f_{n+1} + h \sum_{j=0}^{p} \beta_j f_{n-j}$$

I passi $\mathcal{E}, \mathcal{C}$ possono essere iterati nel seguente modo:

$$\mathcal{E}\, f_{n+1}(k) = f(t_{n+1}, y_{n+1}(k))$$

$$\mathcal{C}\, y_{n+1}(k) = y_n + h\beta_{-1} f_{n+1}(k)$$

dove $f_{n+1}(k)$ è stato calcolato prima.

Si può quindi considerare anche il metodo:

$$\mathcal{P}(\mathcal{E}\mathcal{C})m$$

in cui valutazione e correzione vengono alternate m volte.

Il passo finale è

$$f_{n+1}^m = f(t_{n+1}, y_{n+1}^m)$$

cioè il metodo si prepara per un'eventuale iterazione. Se $\mathcal{P}$ ha ordine $p-1$ i dati di innesco aggiuntivi sono calcolabili con un errore di ordine di $o(h^p)$ e se il *corrector* è di ordine p, allora ($\mathcal{P}\mathcal{C}$) ha ordine p.

Esempio 7.2 (esempio di metodo predictor corrector)

Usiamo come *predictor* l'Eulero esplicito

$$y_{n+1}(0) = y_n + hf_n$$

e come *corrector* il metodo dei trapezi:

$$y_{n+1} = y_n + h/2(f_n + f(t_{n+1}, y_{n+1}(0)))$$

Questo metodo può anche essere considerato come metodo a un passo.

7.4 Metodi Runge-Kutta

7.4.1 Definizione generale

Questi metodi sono del tipo

$$y_{n+1} = y_n + hF(t_n, y_n, h; f)$$

e sono a un passo. F è definita come

$$F(t_n, y_n, h; f) = \sum_{i=1}^{s} b_i * K_i$$

dove i K_i vengono chiamati stadi.

7.4.2 Definizione degli stadi

Gli stadi possono essere definiti come

$$K_i = f(t_n + c_i h, y_n + h \sum_{j=1}^{s} a_{ij} K_j)$$

Nel secondo argomento compaiono nuovamente gli stadi, quindi questi metodi sono non lineari.

I coefficienti c_i, b_i, a_{ij} compaiono nella *batcher array.*

$$\left(\begin{array}{c|ccc} c_1 & a_{11} & \dots & a_{1s} \\ \dots & \dots & \dots & \dots \\ c_s & a_{s1} & \dots & a_{ss} \\ \hline & b_1 & \dots & b_s \end{array}\right)$$

In forma compatta:

$$\begin{pmatrix} \boldsymbol{c} & A \\ & \boldsymbol{b} \end{pmatrix}$$

I coefficienti indicano caratteristiche del metodo usato. In particolare deve valere la *row sum condition*, cioè

$$c_i = \sum_{j=1}^{s} a_{ij}$$

se quest'uguaglianza non è verificata il metodo non funziona.

Si possono verificare tre casi:

1. *metodi espliciti* ($a_{ij} = 0 \iff j \geq i$): la sommatoria si ferma all'indice $i - 1$, la diagonale principale e la parte triangolare superiore di A sono nulle; compare solo la dipendenza dagli stadi precedenti, che sono già stati calcolati;
2. *metodi semi-impliciti* ($a_{ij} = 0 \iff j > i$):la diagonale principale non è nulla, ma la parte triangolare superiore di A lo è, compare la dipendenza dal passo attivo;
3. *metodi impliciti*: A è una matrice piena.

Nel caso dei semi-impliciti dobbiamo risolvere il sistema:

$$K_i = f(t_n + c_i h, y_n + h \sum_{j=1}^{i-1} a_{ij} K_j + h a_{ii} K_i)$$

per $i = 1, \ldots, s$, che è un sistema di s equazioni non lineari disaccoppiate.

Nel caso degli impliciti invece ho un sistema di equazioni non disaccoppiate dove ogni stadio coinvolge tutti gli altri, e le equazioni sono del tipo:

$$K_i = f(t_n + c_i h, y_n) + h \sum_{j=1}^{s} a_{ij} K_j + h a_{ii} K_i$$

7.4.3 Runge-Kutta 4 (esplicito)

Il Runge-Kutta 4 è definito nel seguente modo:

$$y_{n+1} = y_n + h/6(K_1 + 2K_2 + 2K_3 + K_4)$$

dove

$$K_1 = f(t_n, y_n)$$

$$K_2 = f(t_n + h/2, y_n + h/2K_1)$$

$$K_3 = f(t_n + h/2, y_n + h/2K_2)$$

$$K_4 = f(t_{n+1}, y_n + hK_3)$$

Se consideriamo un Runge-Kutta esplicito a s stadi, l'ordine massimo di convergenza è s, anche se si dimostra che non esistono Runge-Kutta espliciti a s stadi di ordine s per $s > 5$.

7.4.4 Consistenza

Un metodo Runge-Kutta è consistente ovvero $\tau_h = \max_n |t_{nh}| \to 0$ per $h \to 0$ se e solo se

$$\sum_{i=1}^{s} b_i = 1$$

In questo caso 1 è l'unica radice del polinomio caratteristico e quindi la consistenza (e in questo caso la 0-stabilità) implica la convergenza.

7.4.5 Runge-Kutta 2

Consideriamo un Runge-Kutta a due stadi, $s = 2$, esplicito. Supponiamo che

$$y_{n+1} = y(t_n) + h(b_1 K_1 + b_2 K_2)$$

con $K_1 = f(t_n, y(t_n))$ e $K_2 = f(t_n + hc_2, y(t_n) + ha_{21} K_1)$.

Il blocco di Batcher è della forma:

$$\left(\begin{array}{ccc} 0 & 0 & 0 \\ c_2 & A_{21} & 0 \\ \hline & b_1 & b_2 \end{array}\right)$$

Deve valere la condizione di somma sulle righe, quindi

$$a_{21} = c_2$$

e si ottiene

$$K_2 = f(t_n + hc_2, y(t_n) + hc_2K_1)$$

e sostituendo l'espressione di K_1:

$$K_2 = f(t_n + hc_2, y(t_n) + hc_2f(t_n, y(t_n))$$

Sviluppando rispetto a h:

$$K_2 = f(t_n, y(t_n)) + c_2hf_t(t_n, y_n) + hc_2f_y(t_n, y(t_n)) + o(H^2)$$

(f_t, f_y indicano le derivate parziali rispetto a t e a y) e sostituendo le espressioni degli stadi nell'espressione del metodo:

$$y_{n+1} = y(t_n)+hb_1f(t_n, y(t_n))+hb_2[f(t_n, y(t_n))+hc_2f_t(t_n, y(tn))+hc_2f_y(t_n, y(t_n)+O(H^2))$$

e associando i termini corrispondenti

$$y_{n+1} = y(t_n) + h(b_1 + b_2)f(t_n, y(t_n)) + h^2b_2c_2(f_t(t_n, y(tn)) + f_y(t_n, y(t_n)) + O(H^2)$$

ma $y_{n+1} = y(t_n) + hy'(t_n) + h^2/2y''(t_n) + O(h^3)$, quindi

$$y(t_n) + hy'(t_n) + h^2/2y''(t_n) + O(h^3) =$$

$$= y(t_n) + h(b_1 + b_2)f(t_n, y(t_n)) + h^2b_2c_2(f_t(t_n, y(tn)) + f_y(t_n, y(t_n)) + O(H^2)$$

e

$$hy'(t_n) + h^2/2y''(t_n) + O(h^3) =$$

$$= h(b_1 + b_2)f(t_n, y(t_n)) + h^2b_2c_2(f_t(t_n, y(tn)) + f_y(t_n, y(t_n)) + O(H^2)$$

Imponiamo che la soluzione esatta soddisfi lo schema numerico fino a un certo ordine, e confrontando i coefficienti a primo e secondo membro otteniamo:

$$\begin{cases} b_1 + b_2 = 1 \\ b_2c_2 = 1/2 \end{cases}$$

Ci sono infinite soluzioni possibili, quindi infiniti Runge-Kutta di tipo 2.

7.4.6 Passo di integrazione

Dato un problema di integrazione *stiff*, si deve sempre scegliere un passo di integrazione variabile. Nei metodi Runge-Kutta a un passo il passo può essere variato a piacere, mentre nei metodi multipasso questo non è vero.

I Runge-Kutta impliciti sono i metodi che raggiungono la massima potenza possibile. La loro teoria di convergenza viene trattata in maniera algebrica.

7.5 Analisi di stabilità lineare o assoluta stabilità

Nella pratica, $h \not\to 0$, quindi la 0-stabilità non basta.

Consideriamo il problema lineare:

$$\begin{cases} y' = \lambda y \\ y(0) = 1 \end{cases}$$

con $\lambda \in \mathbb{C}$. La soluzione esatta è

$$y(t) = e^{\lambda t}$$

ed è tale che

$$\lim_{t \to \infty} |y(t)| = 0$$

se $Re(\lambda) < 0$.

Verifichiamo la seguente definizione:

◊ **Definizione** Un **metodo numerico** è **assolutamente stabile** se

$$|y_{n+1}| \to 0 \iff n \to \infty$$

(quindi se la soluzione numerica si comporta come la soluzione esatta).

◊ **Definizione** Si definisce la **regione di assoluta stabilità**

$$\mathcal{A} = \{z = \lambda h \in \mathbb{C}\, t.c. \, |y_{n+1}| \to 0 \iff n \to \infty\}$$

Supponiamo che esistano due costanti μ_{min}, μ_{max} tali che $-\mu_{min} < f_y < -\mu_{max}$, e poniamo $\lambda = -\mu_{max}$.

La regione di assoluta stabilità varia a seconda del metodo considerato, infatti:

1. considerando il metodo di Eulero esplicito:

$$y_{n+1} = y_n + hf_n$$

se $f = \lambda y$ segue che

$$y_{n+1} = y_n + \lambda h y_n = (1 + \lambda h) y_n = (1 + \lambda h)^n y_0$$

e per $n \to \infty$ la quantità tende a 0 se $|1 + \lambda h| < 1$. Il metodo di Eulero esplicito ha come condizione di assoluta stabilità $|1 + \lambda h| < 1$. Nel piano di Argand-Gauss la

regione di assoluta stabilità è un cerchio con raggio 1 e centro 1. λh deve stare nel cerchio, altrimenti asintoticamente la soluzione numerica non si comporta come la soluzione esatta. In $\mathbb{R}$ la condizione diventa $h < -2/\lambda$.

2. considerando invece il metodo di Eulero implicito:

$$y_{n+1} = y_n + hf_{n+1} = y_n + h\lambda y_{n+1}$$

$$(1 - \lambda h)y_{n+1} = y_n$$

$$y_{n+1} = \frac{y_n}{1 - \lambda h}$$

e ricorsivamente

$$y_{n+1} = \frac{y_0}{(1 - \lambda h)^n}$$

e imponiamo la condizione $|\frac{1}{1-\lambda h}| < 1$ che implica $|1 - \lambda h| > 1$. La regione di assoluta stabilità in questo caso è ciò che sta fuori dal cerchio di raggio 1 e centro 1. Quindi i metodi di Eulero implicito non hanno nessuna condizione sulla scelta del passo dovuti allo schema numerico.

3. Se consideriamo il metodo di Frank-Nicholson,

$$y_{n+1} = y_n + h/2(f_n + f_{n+1})$$

segue che, se $f = \lambda y$, si ha

$$y_{n+1} = y_n + h/2\lambda y_n + h/2\lambda y_{n+1}$$

$$(1 - h/2\lambda)y_{n+1} = y_n + h/2\lambda y_n = (1 + h/2\lambda)y_n = (1 + h/2\lambda)^n y_0$$

quindi

$$y_{n+1} = \frac{(1 + h/2\lambda)^n}{1 - h/2\lambda} y_0$$

e si ottiene la condizione

$$|\frac{1 + h/2\lambda}{1 - h/2\lambda}| = |\frac{2 + h\lambda}{2 - h\lambda}| < 1$$

$$|2 + h\lambda| < 2 - h\lambda$$

$$h\lambda - 2 < 2 + h\lambda < 2 - h\lambda$$

$$2h\lambda < 0$$

7.5.1 Trasformazione di sistemi lineari

Consideriamo il problema di Cauchy

$$\begin{cases} y' = F(t, y) \\ y(t_0) = y_0 \end{cases}$$

con $F\colon \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$. È un sistema di equazioni differenziali con

$$y' = Ay$$

Assumiamo che $A = S^{-1}\lambda S$ sia diagonalizzabile.

La soluzione esatta è

$$y(t) = \sum_{j=1}^{n} c_j e^{\lambda_j t} b_j$$

$y(t) \to 0 \iff t \to \infty$, e si richiede lo stesso comportamento per la soluzione numerica.

Allora si impone:

$$z = S^{-1}y \longrightarrow z' = S^{-1}y'$$

$$y' = Sz'$$

$$y' = Ay \longrightarrow Sz' = ASz$$

quindi, moltiplicando a sinistra per S^{-1}

$$z' = S^{-1}ASz = \lambda z$$

si ha un sistema di equazioni differenziali lineari ma con λ diagonale, in modo da avere equazioni differenziali disaccoppiate della forma:

$$z_i' = \lambda_i z_i$$

e consideriamo il problema test su ciascuna delle equazioni. Scegliamo come passo h, che deve valere per tutte le equazioni, l'autovalore di modulo maggiore.

7.5.2 Problemi *stiff*

Supponiamo di avere un sistema di equazioni differenziali del tipo

$$\left\{ y' = A\phi(x) \right.$$

con $\phi \in \mathbb{R}^n$. Supponiamo che A sia diagonalizzabile, e consideriamo il caso in cui le parti reali dei suoi autovalori siano negative.

Allora la soluzione esatta è nella forma

$$y(t) = \sum_{j=1}^{n} c_j e^{\lambda_j t} v_j + \psi(t)$$

dove $\psi(t)$ è La soluzione particolare della non omogenea.

Considerando un metodo numerico a una regione di assoluta stabilità limitata, allora ci sono necessariamente delle restrizioni sul passo di integrazione. Più piccola è la regione di assoluta stabilità, minore è il passo da scegliere, e la scelta del passo è dovuta alla soluzione transitoria. Il passo va scelto tanto più piccolo quanto più in fretta la soluzione tende a 0.

Chiamiamo λ_m, λ_M due autovalori della matrice A tali che valga la relazione:

$$|\,Re(\lambda_M)| \geq |\,Re(\lambda_j)| \geq Re(|\lambda_m|)$$

Si definisce la *stiffness ratio* nel seguente modo:

$$\frac{|\,Re(\lambda_M)|}{|\,Re(\lambda_m)|}$$

e il problema è *stiff* se questa quantità è molto maggiore di 1.

Se $Re(\lambda_m) \sim 0$, anche se $Re(\lambda_M)$ non è molto grande, abbiamo un problema *stiff* per definizione. I metodi utilizzati per risolverli sono i Runge-Kutta espliciti.

Crediti

Fonti dei testi

- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Introduzione_all%27aritmetica_floating_point&oldid=473864
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Cos%27%C3%A8_l%27aritmetica_floating_point&oldid=473889
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Rappresentazione_floating_dei_numeri&oldid=473900
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Operazioni_aritmetiche_con_il_calcolatore&oldid=473654
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Teoria_dell%27errore_nella_sua_generalit%C3%A0&oldid=473903
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Condizionamento_e_stabilit%C3%A0&oldid=473888
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Comportamento_dell%27errore_algoritmico_e_dell%27errore_analitico&oldid=473887
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Norme_di_vettori_e_di_matrici&oldid=473874
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Sistemi_lineari_e_errore_inerente_(condizionamento)&oldid=473762
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Sistemi_lineari_su_un_calcolatore&oldid=473901
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Fattorizzazione_LU_(eliminazione_gaussiana)&oldid=473858
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Tecniche_del_pivot&oldid=473837
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Fattorizzazione_di_Cholesky&oldid=473857
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Fattorizzazione_QR&oldid=473890
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Metodi_iterativi&oldid=473904
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Principali_metodi_iterativi&oldid=473906
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Metodi_di_rilassamento&oldid=473907
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Localizzazione_degli_autovalori&oldid=473928

- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Calcolo_di_autovalori_estremi&oldid=473936
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Metodo_della_potenza&oldid=473942
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Metodo_della_potenza_inversa&oldid=473944
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Casi_particolari&oldid=473945
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Radici_di_equazioni_non_lineari&oldid=473947
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Bisezione_corde_secanti_Newton&oldid=473948
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Metodo_di_Brent&oldid=473950
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Metodi_del_punto_fisso&oldid=473953
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Condizionamento_test_e_stabilit%C3%A0_dei_metodi_di_punto_fisso&oldid=473955
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Radici_di_sistemi_non_lineari&oldid=473956
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Interpolazione_polinomiale&oldid=474531
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Altre_scritture_dei_polinomi_interpolanti&oldid=474064
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Stabilit%C3%A0_condizionamento_e_convergenza&oldid=474066
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Interpolazione_polinomiale_a_tratti_(spline)&oldid=474098
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Approssimazione_media_dei_minimi_quadrati&oldid=474099
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Approssimazione_dei_minimi_quadrati_con_funzioni_continue&oldid=474100
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Formule_di_quadratura_approssimate&oldid=474101
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Formule_di_Newton-Cotes&oldid=474182
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Integrazione_automatica&oldid=474190
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Problemi_di_Cauchy&oldid=474192
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Algoritmi_per_la_risoluzione&oldid=474198
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Metodi_multipasso&oldid=474533
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Metodi_Runge-Kutta&oldid=474534
- https://it.wikibooks.org/w/index.php?title=Calcolo_numerico/Analisi_di_stabilit%C3%A0_lineare_o_assoluta_stabilit%C3%A0&oldid=474204