Dispense di Calcolo numerico

Emanuele Izzo, 0307385

Corso di laurea triennale Informatica
Università Tor Vergata, Facoltà di Scienze MM.FF.NN.
06/08/2020

Documento realizzato in LaTeX

Indice

1 Introduzione **3**
- 1.1 Analisi numerica . . . 3
- 1.2 Teoria degli errori . . . 3
- 1.3 Errore assoluto ed errore relativo . . . 4
- 1.4 Tipologie di errori . . . 4
- 1.5 Problema bencondizionato . . . 4
- 1.6 Algoritmo stabile . . . 5
- 1.7 Stima del condizionamento del calcolo di $f(x)$ . . . 5

2 Rappresentazione dei numeri **6**
- 2.1 Rappresentazione normalizzata base β . . . 6
- 2.2 Numeri macchina . . . 6
- 2.3 Arrotondamento e troncamento . . . 7
- 2.4 Limitazione superiore degli errori assoluti . . . 7
- 2.5 Limitazione superiore degli errori relativi . . . 7

3 Aritmetica finita **8**
- 3.1 Propagazione degli errori (analisi in avanti del primo ordine) . . . 8
- 3.2 Precisione di macchina (definizione alternativa) . . . 10

4 Equazioni non lineari **11**
- 4.1 Procedimenti iterativi . . . 11
- 4.2 Convergenza locale e globale . . . 12
- 4.3 Ordine e fattore di convergenza . . . 12
- 4.4 Metodo di bisezione . . . 12
- 4.5 Metodo della regula falsi . . . 13
- 4.6 Versione linearizzata . . . 14
- 4.7 Il metodo delle secanti . . . 14
- 4.8 Metodo di Newton . . . 15
- 4.9 Metodi di iterazione funzionale . . . 15
- 4.10 Convergenza . . . 17
  - 4.10.1 Teorema di Ostrowski . . . 17
- 4.11 Condizione di convergenza o divergenza locale . . . 18
- 4.12 Ordine di convergenza . . . 18
  - 4.12.1 Teorema . . . 19
- 4.13 Convergenza del metodo di Newton . . . 19
  - 4.13.1 Teorema . . . 19

5 Richiami di algebra lineare **20**
5.1 Proprietà delle matrici . . . 20
5.1.1 Teorema . . . 20
5.2 Norma di vettore . . . 21
5.3 Norme equivalenti . . . 21
5.4 Norma di matrici compatibile o consistente . . . 21
5.5 Norma di matrici indotta . . . 21
5.6 Norma di matrici naturale . . . 22
5.7 Principali norme indotte naturali . . . 22
5.8 Proprieà delle norme naturali . . . 22

6 Cerchi di Gerschgorin e autovalori **23**
6.1 Primo teorema di Gerschgorin . . . 23
6.2 Secondo teorema di Gerschgorin . . . 23
6.3 Terzo teorema di Gerschgorin . . . 23
6.4 Proprietà . . . 23

7 Sistemi lineari quadrati **25**
7.1 Condizionameto . . . 25
7.1.1 Teorema . . . 26
7.2 Metodo di Cramer . . . 26
7.3 Algoritmo di sostituzione in avanti . . . 26
7.4 Algoritmo di sostituzione all'indietro . . . 27
7.5 Fattorizzazione LR . . . 28
7.6 Teorema di fattorizzazione . . . 28
7.6.1 Teorema . . . 28
7.7 Fattorizzazione LR per matrici simmetriche . . . 28
7.8 Fattorizzazione di Choleski . . . 29
7.9 Metodo di eliminazione di Gauss . . . 30
7.10 Metodo di Gauss con pivoting e scaling . . . 32
7.10.1 Teorema . . . 34
7.11 Metodi iterativi . . . 34
7.12 Metodo iterativo convergente . . . 35
7.12.1 Teorema . . . 35
7.13 Velocità di convergenza . . . 35
7.13.1 Teorema . . . 36
7.13.2 Teorema . . . 36
7.13.3 Teorema . . . 36
7.14 Teorema di Stein-Rosemberg . . . 36

8 Interpolazione **37**
8.1 L'interpolazione polinomiale . . . 38
8.2 Polinomio interpolante di Lagrange . . . 38
8.3 Polinomio interpolante di Newton . . . 39

9 Errore di interpolazione **41**

Capitolo 1

Introduzione

1.1 Analisi numerica

L'analisi numerica è lo studio degli alogritmi per la soluzione di problemi della matematica del continuo [*N. Trefethen*]. L'obiettivo è quello di dare una risposta numerica ad un problema matematico mediante un calcolatore automatico digitale, tramite 5 passaggi:

1. Problema reale;
2. Costruzione del modello;
3. Formulazione di un problema matematico;
4. Risoluzione del problema matematico;
5. Interpretazione della soluzione.

1.2 Teoria degli errori

In ognuno dei passi 2, 3 e 4 si introducono degli errori:

2-3) Ipotesi semplificative nella costruzione del modello;

4.1) Errori di misura ed errore dei rappresentazione dei dati;

4.2) Errori di troncamento : approssimazione di un processo infinito con un finito di calcolo effettivo;

4.3) Errori di arrotondamento nei calcoli: esecuzione effettiva dell'algoritmo.

É importante che gli errori introdotti nei vari stadi siano dello stesso ordine. É inutile calcolare accuratamente la soluzione di un problema matematico derivante da un modellazione non accurata. L'analisi numerica si occupa essenzialmente della fase 4.

1.3 Errore assoluto ed errore relativo

Definizione: Data una grandezza (scalare) x ed una sua approssimazione $\tilde{x}$ definiamo:

- $\delta_x = |\tilde{x} - x|$ **errore assoluto** ($\tilde{x} = x + \delta_x$);
- $\epsilon_x = \dfrac{|\tilde{x} - x|}{|x|}$ **errore relativo** ($\tilde{x} = x + x\epsilon_x = x(1 + \epsilon_x)$);

1.4 Tipologie di errori

- I dati x sono affetti da un errore (di misura o di rappresentazione dovuta al calcolatore) δ_x [*errore inerente A*]:

$$\frac{|f(x + \delta_x) - f(x)|}{|f(x)|}$$

- Anzichè calcolare f, si calcola una sua approssimazione g (numero finito di passi) [*errore analitico o di troncamento B*]:

$$\frac{|g(x) - f(x)|}{|f(x)|}$$

- Il calcolo di g viene effettuato con l'esecuzione di uno specifico algoritmo su un calcolatore (precisione finita) ottenendo un'approssimazione $\tilde{g}$ [*errore algoritmico C*]:

$$\frac{|\tilde{g}(x) - g(x)|}{|g(x)|}$$

Alla fine, invece di $y = f(x)$, si ottiene $\tilde{y} = \tilde{g}(x + \delta_x)$.

1.5 Problema bencondizionato

Sappiamo che gli errori A dipendono dal problema e dal dato, e non dall'algoritmo utilizzato. Da qui possiamo dare una definizione di problema bencondizionato.

Definizione: Un problema è **bencondizionato** se, a piccole variazioni sui dati, corrispondono piccole variazioni sui risultati, ossia se

$$\left|\frac{f(x + \delta_x) - f(x)}{f(x)}\right| = \left|\frac{\delta_x}{x}\right| = \epsilon_x$$

1.6 Algoritmo stabile

Sappiamo che gli errori C dipendono dalle caratteristiche dell'algoritmo utilizzato (operazioni e precisione) ed è dovuto all'uso dell'aritmetica finita. Da qui possiamo dare una definizione di algoritmo stabile.

Definizione: Un algoritmo è **stabile** se amplifica poco (relativamente alle caratteristiche del calcolatore) gli errori di arrotondamento introdotti nelle singole operazioni.

Possiamo fare alcune osservazioni a rigardo:

- Un problema può essere malcondizionato per certi dati e per altri no;
- Anche un problema bencondizionato può dare risultati non sufficientemente corretti se risolto con un algoritmo instabile;
- La stabilità di un algoritmo è data dal bencondizionamento della successione delle trasformazioni elementari che la compongono.

1.7 Stima del condizionamento del calcolo di $f(x)$

Data una grandezza (scalare) x e il suo errore relativo ϵ_x definiamo

$$\frac{f(x + x\epsilon_x) - f(x)}{f(x)} = \frac{f'(\xi)(x + x\epsilon_x - x)}{f(x)} = \frac{f'(x)x}{f(x)} * \epsilon_x = C_x\epsilon_x$$

Dove $C_x = \dfrac{f'(x)x}{f(x)}$ è il coefficiente di amplificazione dell'errore dei dati. Due passaggi chiave:

- $F(b) - F(a) = f(\xi)(b - a)$ con $b > a$, $\xi \in [a, b]$ e $F(x)$ continua in $[a, b]$ e derivabile in (a, b);
- Essendo $\xi \in [a, b]$, possiamo dire che $\xi \tilde{=} x$ (ciò è possibile dato che ϵ_x è "piccolo").

Capitolo 2

Rappresentazione dei numeri

2.1 Rappresentazione normalizzata base β

Sia data una base $\beta \geq 2$, sia $x \in \mathbb{R}$ con $x \neq 0$, esiste uno ed un solo intero $k \in \mathbb{Z}$ ed una successione $\{d_i\}_{i\in[1,n]}$ con:

1. $0 \geq d_i \geq \beta - 1$, $d_1 \neq 0$;
2. $\{d_i\}_{i\in[1,n]}$ non definitivamente uguali a $\beta - 1$ (ossia $\exists i \in [1, n] | d_i \neq \beta - 1$);

tale che $x = \text{sign}(x)\beta^k \sum_{i=1}^{n} d_i\beta^{-i}$, dove:

- k è l'esponente o caratteristica;
- d_i sono le cifre della rappresentazione;
- $\sum_{i=1}^{n} d_i\beta^{-i}$ è la mantissa.

Questa rappresentazione è detta rappresentazione in virgola mobile.

2.2 Numeri macchina

L'insieme dei numeri rappresentabili in un calcolatore è dato da F(β, t, L, U), dove:

- β è la base;
- t sono le cifre di mantissa;
- L è il limite inferiore;
- U è il limite superiore.

e l'intervallo di rappresentazione è $x \in$F(β, t, L, U)$\subset [\beta^{\mathrm{L}-1}, (1 - \beta^{-t})\beta^{\mathrm{U}}]$.

2.3 Arrotondamento e troncamento

Sia $x = (0, d_i d_2 ... d_n)_\beta \beta^k$ tale che, per la macchina F(β, t, L, U), $x \notin$F (in particolare L$\geq k \geq$U ma $n > t$), allora è possibile effettuare una di queste due operazioni:

- Troncamento: trunc$(x) = (0, d_i d_2 ... d_t)_\beta \beta^k$ dove $x = (0, d_i d_2 ... d_t ... d_n)_\beta \beta^k$;
- Arrotondamento: arr$(x) = (0, d_i d_2 ... (d_t + 1))_\beta \beta^k$ se $d_{t+1} > \beta/2$, oppure arr$(x) = (0, d_i d_2 ... d_t)_\beta \beta^k$ se $d_{t+1} < \beta/2$ (in caso in cui $\beta_{t+1} = d_{t+1}$ si può scegliere come agire, ossia se arrotondare, troncare o rendere pari d_t se non lo è).

2.4 Limitazione superiore degli errori assoluti

Data una grandezza (scalare) x definiamo:

- Per troncamento: $|\text{trunc}(x) - x| < \beta^{k-t}$;
- Per arrotondamento $|\text{arr}(x) - x| < \frac{1}{2}\beta^{k-t}$.

2.5 Limitazione superiore degli errori relativi

Data una grandezza (scalare) x definiamo:

- Per troncamento: $\frac{|\text{trunc}(x) - x|}{|x|} < \frac{\beta^{k-t}}{\beta^{k-1}} = \beta^{1-t} = \epsilon_m$;
- Per arrotondamento: $\frac{|\text{arr}(x) - x|}{|x|} < \frac{1}{2}\frac{\beta^{k-t}}{\beta^{k-1}} = \frac{1}{2}\beta^{1-t} = \frac{1}{2}\epsilon_m$.

dove $\epsilon_m = \beta^{1-t}$ è detta precisione di macchina. Da qui abbiamo che, definendo con $fl(x)$ una generica funzione di troncamento/arrotondamento, $fl(x) = x(1 + \epsilon)$, $|\epsilon| < \epsilon_m$.

Capitolo 3

Aritmetica finita

Dati due numeri macchina x e y, non è detto che $x(\text{op})y$ sia acnora un numero macchina, allora occorre definire un'aritmetica approssimata o aritmetica finita: x (op) $y = fl(x \text{ op } y) = (x \text{ op } y)(1+\epsilon)$, $|\epsilon| < \epsilon_m$. Un'aritmetica basata sull'arrotondamento è più precisa, ma necessita, per essere eseguita, di registri più lunghi (per esaminare la $(t+1)$-esima cifra) e quindi di più tempo.

- $x + y \to x \oplus y = fl(fl(x) + fl(y))$
- $x - y \to x \ominus y = fl(fl(x) - fl(y))$
- $x * y \to x \otimes y = fl(fl(x) * fl(y))$
- $x/y \to x \oslash y = fl(fl(x)/fl(y))$

Proprietà

Dati tre numeri macchina x, y e z, abbiamo che

- $x \oplus y = y \oplus x$;
- $x \otimes y = y \otimes x$;
- $(x \oplus y) \oplus z \neq x \oplus (y \oplus z)$;
- $(x \otimes y) \otimes z \neq x \otimes (y \otimes z)$;
- $x \otimes (y \oplus z) \neq (x \otimes y) \oplus (x \otimes z)$;
- $x \oplus y = x \not\oplus y = 0$;
- $x \otimes (y \oslash x) \neq y$.

3.1 Propagazione degli errori (analisi in avanti del primo ordine)

Date due grandezze (scalari) x ed y e una loro approssimazione:

- $\tilde{x} = x + \delta_x = x(1 + \epsilon_x)$, $|\epsilon_x| < \epsilon_m$;

- $\tilde{y} = y + \delta_y = y(1 + \epsilon_y)$, $|\epsilon_y| < \epsilon_m$.

abbiamo che:

- $x \oplus y = (x + y)(1 + \epsilon_s)$

$$\frac{\tilde{x} \oplus \tilde{y} - (x + y)}{x + y} = \frac{(\tilde{x} + \tilde{y})(1 + \epsilon_s) - (x + y)}{x + y} =$$
$$= \frac{x(1 + \epsilon_x)(1 + \epsilon_s) + y(1 + \epsilon_y)(1 + \epsilon_s) - (x + y)}{x + y} =$$
$$= \frac{x + x\epsilon_s + x\epsilon_x + x\epsilon_x\epsilon_s + y + y\epsilon_y + y\epsilon_s + y\epsilon_y\epsilon_s - x - y}{x + y} =$$
$$= \frac{x}{x + y}\epsilon_x + \frac{y}{x + y}\epsilon_y + \epsilon_s$$

  e quindi si hanno problemi se $|x| \cong |y|$ ma $x = -y$ (i valori $\epsilon_x\epsilon_s \cong \epsilon_m^2$ e $\epsilon_y\epsilon_s \cong \epsilon_m^2$ sono trascurabili);

- $x \ominus y = (x - y)(1 + \epsilon_s)$

$$\frac{\tilde{x} \ominus \tilde{y} - (x - y)}{x - y} = \frac{(\tilde{x} - \tilde{y})(1 + \epsilon_s) - (x - y)}{x - y} =$$
$$= \frac{x(1 + \epsilon_x)(1 + \epsilon_s) - y(1 + \epsilon_y)(1 + \epsilon_s) - (x - y)}{x - y} =$$
$$= \frac{x + x\epsilon_s + x\epsilon_x + x\epsilon_x\epsilon_s - y - y\epsilon_y - y\epsilon_s - y\epsilon_y\epsilon_s - x + y}{x - y} =$$
$$= \frac{x}{x - y}\epsilon_x - \frac{y}{x - y}\epsilon_y + \epsilon_s$$

  e quindi si hanno problemi se $x \cong y$ (i valori $\epsilon_x\epsilon_s \cong \epsilon_m^2$ e $\epsilon_y\epsilon_s \cong \epsilon_m^2$ sono trascurabili);

- $x \otimes y = (x * y)(1 + \epsilon_s)$

$$\frac{\tilde{x} \otimes \tilde{y} - (x * y)}{x * y} = \frac{(\tilde{x} * \tilde{y})(1 + \epsilon_s) - (x * y)}{x * y} =$$
$$= \frac{x(1 + \epsilon_x)y(1 + \epsilon_y)(1 + \epsilon_s) - (x * y)}{(x * y)} =$$
$$= \frac{xy + xy\epsilon_x + xy\epsilon_y + xy\epsilon_x\epsilon_y + xy\epsilon_s + xy\epsilon_x\epsilon_s + xy\epsilon_y\epsilon_s + xy\epsilon_x\epsilon_y\epsilon_s - xy}{xy} =$$
$$= \frac{xy(1 + \epsilon_x + \epsilon_y + \epsilon_s - 1)}{xy} = \epsilon_x + \epsilon_y + \epsilon_s$$

  (i valori $\epsilon_x\epsilon_s \cong \epsilon_m^2$, $\epsilon_y\epsilon_s \cong \epsilon_m^2$ e $\epsilon_x\epsilon_y\epsilon_s \cong \epsilon_m^3$ sono trascurabili);

- $x \oslash y = (x/y)(1 + \epsilon_s)$

$$\frac{\tilde{x} \oslash \tilde{y} - (x/y)}{x/y} = \frac{(\tilde{x}/\tilde{y})(1 + \epsilon_s) - (x/y)}{x/y} =$$
$$= \frac{\dfrac{x(1 + \epsilon_x)}{y(1 + \epsilon_y)}(1 + \epsilon_s) - \dfrac{x}{y}}{x/y} = \frac{x}{y}\left(\frac{1 + \epsilon_x}{1 + \epsilon_y}(1 + \epsilon_s)\right)\frac{y}{x} =$$
$$= \frac{1 + \epsilon_x + \epsilon_s + \epsilon_x\epsilon_s}{1 + \epsilon_y} = \frac{1 + \epsilon_x + \epsilon_s}{1 + \epsilon_y}$$

  (il valore $\epsilon_x\epsilon_s \cong \epsilon_m^2$ è trascurabile).

3.2 Precisione di macchina (definizione alternativa)

Si definisce precisione di macchina il valore
$\epsilon_m \in \mathbb{R} | 1 \oplus \epsilon_m > 1 \wedge \forall \epsilon \in \mathbb{R} : |\epsilon| < |\epsilon_m|, 1 \oplus \epsilon = 1$.

Dimostrazione

Sia F(β, t, L, U) una macchina che usa trunc, calcoliamo i valori:

- $\epsilon_m = \beta^{1-t} = 0,1\beta^{2-t}$;
- $1 = 0,1\beta^1$;
- $\epsilon = \beta^{1-t}\beta^{-1} = 0,1\beta^{1-t}$, in modo che $|\epsilon| < |\epsilon_m|$.

calcoliamo:

- $1 \oplus \epsilon_m = 0,1\beta^1 + 0,\underbrace{0000...0}_{t-1 \text{ volte}}1 = 0,\overbrace{1\underbrace{0000...0}_{t-2 \text{ volte}}1}^{t \text{ cifre}}\beta^1 > 1$;
- $1 \oplus \epsilon = 0,1\beta^1 + 0,\underbrace{0000...0}_{t \text{ volte}}1 = 0,\overbrace{1\underbrace{0000...0}_{t-1 \text{ volte}}1}^{t+1 \text{ cifre}}\beta^1$, ma
  $0,\underbrace{10000...01}_{t+1 \text{ cifre}}\beta^1 \notin F$, quindi $\text{trunc}(0,\overbrace{\underbrace{10000...0}_{t+1 \text{ cifre}}1}^{t \text{ cifre}}\beta^1) = 1$.

Capitolo 4

Equazioni non lineari

Data una funzione $f : \mathbb{R} \to \mathbb{R}$ consideriamo il problema di determinare i valori x tali che $f(x) = 0$. Tali valori sono solitamente chiamati zeri o radici della funzione f. In generale non sono disponibili formule esplicite per la determinazione delle radici di una funzione. Per effettuare ciò vengono usati dei metodi iterativi, ossia delle tecniche che consentono di approssimare le soluzioni con un prestabilito grado di precisione. A partire de una approssimazione iniziale x_0 si costruisce una successione $x_1, x_2, ..., x_n$ che, sotto opportune ipotesi, risulta convergere alla radice cercata.

4.1 Procedimenti iterativi

Sia P un problema ed α una soluzione del problema P. Supponiamo di utilizzare un procedimento iterativo per la determinazione di α che genera una successione $\{x_i\}_{i\in[1,n]}$ convergente ad α, è importante tener presente tre questioni fondamentali:

- Scelta del valore di innesco e convergenza: supponiamo di considerare procedimenti iterativi ricorrenti ad un passo $x_{i+1} = \phi(x_i)$, è necessario per poter innescare il procedimento un punto di innesco x_0;
- Velocità di convergenza;
- Criteri di arresto: chiaramente non è possibile generare infinite iterate della successione, infatti il procedimento dovrebbe arrestarsi quando $|e_i| = |x_i - \alpha| < toll$; non disponendo della soluzione è necessario procurarsi una stima di e_i, ed una possibile strategia è quella di approssimare e_i con $|x_{i+1} - x_i|$, ottenendo così il criterio di arresto assoluto $|x_{i+1} - x_i| < toll_A$; il criterio di arresto assoluto può fallire se la tolleranza è troppo alta grande o piccola, e conviene pertanto usare il criterio di arresto relativo $\frac{|x_{i+1} - x_i|}{|x_{i+1}|} < toll_R$ (**Osservazione**: la tolleranza utilizzata nel criterio di arresto relativo non deve essere minore della precisione di macchina, ma $toll_R > \epsilon_m$); per equazioni non lineari si può anche usare il controllo del residui $|f(x_{i+1})| < toll$, dove:
  - Se $|f(\alpha)| << 1$ è inaffidabile;
  - Se $|f(\alpha)| >> 1$ è troppo restrittivo;
  - Se $|f(\alpha)| \cong 1$ produce un'indicazione soddisfacente.

quando non si ha la certezza della bontà del test (su x e su f), conviene includerle entrambi; in pratica, se il criterio di arresto funziona, non si ha la soluzione α, ma solo una sua approssimazione.

4.2 Convergenza locale e globale

Un metodo converge localmente ad α se la convergenza della successione $\{x_i\}$ dipende in modo critico dalla vicinanza di x_0 ad α. Il procedimento è globalmente convergente quando la convergenza non dipende da quanto x_0 è vicino ad α.

4.3 Ordine e fattore di convergenza

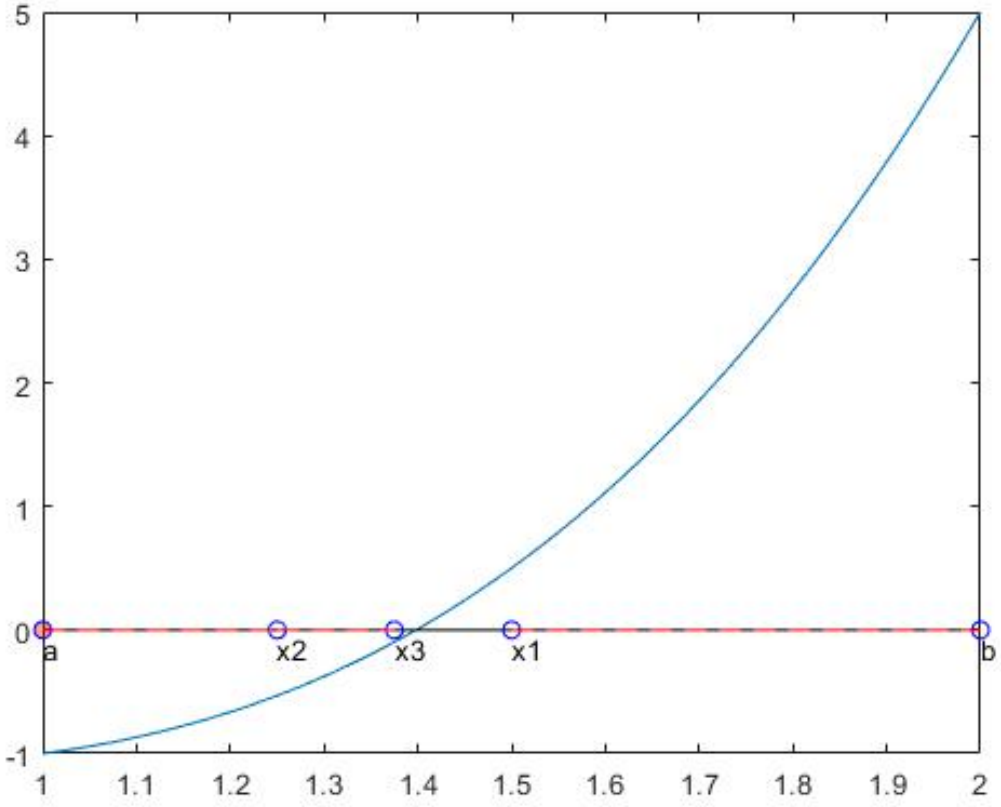
Data una successione $\{x_i\}$ convergente ad un limite α, si ponga $e_i = x_i - \alpha$. Se esistono due numeri reali p e C tali che sia $\lim_{i\to\infty} \frac{|e_{i+1}|}{|e_i|^p} = \mathrm{C}$ si dice che la successione ha ordine di convergenza p e fattore di convergenza C. Per $p = 1$ la convergenza si dice lineare. Per $p = 2$ la convergenza si dice quadratica. Nel caso $p = 1$ si deve necessariamente avere $\mathrm{C} < 1$. Un metodo iterativo è convergente di ordine p se tale è la successione da esso generata.

4.4 Metodo di bisezione

Il metodo di bisezione è il metodo iterativo più semplice per approssimare gli zeri di una funzione. Se $f(x)$ è continua nell'intervallo $[a, b]$ e se $f(a)f(b) < 0$, allora, per il teorema degli zeri, ammette almeno una soluzione α di $f(x) = 0$ in (a, b). Si provvede suddividendo ad ogni passo l'intervallo $[a, b]$ a metà e determinando in quale dei due sottointervalli si trova la soluzione, dimezzando così l'ampiezza dell'intervallo che contiene α.

Algoritmo

1. Si pone $a_0 = a$ e $b_0 = b$;
2. $\forall i \in [1, n]$, calcolo $x_i = \frac{a_{i-1} + b_{i-1}}{2}$;
3. Effettuo tre controlli:
   (a) Se $f(x_i)f(a_{i-1}) < 0$ allora $a_i = a_{i-1}$ e $b_i = x_i$;
   (b) Se $f(x_i)f(b_{i-1}) < 0$ allora $a_i = x_i$ e $b_i = b_{i-1}$;
   (c) Se $f(x_i) = 0$ allora $\alpha = x_i$.
4. Il procedimento viene arrestato se per un indice i risulta $|f(x_i)| \leq toll$ o $|a_i - b_i| \leq toll$.



Proprietà

Il metodo di bisezione converge globalmente alla soluzione con la sola ipotesi che f sia continua nell'intervallo $[a, b]$. La convergenza è però lenta e questo costituisce il limite del metodo: ad ogni passo si riduce l'errore di 1/2 (per ridurlo di di 1/10 occorrono circa 3,3 passi). Una spiegazione può essere ricercata nel fatto che non si tiene conto dei valori della funzione ma soltanto dei segni. Geometricamente il metodo costruisce ad ogni passo l'approssimazione della radice calcolando l'intersezione con le ascisse della retta passante per i punti $(a, \text{sign}(f(a)))$ e $(b, \text{sign}(f(b)))$.

4.5 Metodo della regula falsi

Un modo naturale per migliorare il metodo di bisezione è quello di considerare anche i valori che la funzione assume negli estremi dell'intervallo. Si prende come nuova approssimazione della soluzione l'intersezione delle ascisse con la retta passante per $(a, f(a))$, $(b, f(b))$, ossia

$$\begin{cases} y - f(a) = \dfrac{f(b) - f(a)}{b - a}(x - a) \\ y = 0 \end{cases}$$

da cui si ottiene che $x = a - f(a)\dfrac{b - a}{f(b) - f(a)}$. Il metodo risultante è noto come metodo della regula falsi o della posizione.

Algoritmo

1. Si pone $a_0 = a$ e $b_0 = b$;
2. $\forall i \in [1, n]$, calcolo $x_i = a_{i-1} - f(a_{i-1})\dfrac{b_{i-1} - a_{i-1}}{f(b_{i-1}) - f(a_{i-1})}$
3. Effettuo tre controlli:
   (a) Se $f(x_i)f(a_{i-1}) < 0$ allora $a_i = a_{i-1}$ e $b_i = x_i$;
   (b) Se $f(x_i)f(b_{i-1}) < 0$ allora $a_i = x_i$ e $b_i = b_{i-1}$;
   (c) Se $f(x_i) = 0$ allora $\alpha = x_i$.
4. Il procedimento viene arrestato se per un indice i risulta $|f(x_i)| \leq toll$ o $|a_i - b_i| \leq toll$.

Proprietà

Il metodo genera una successione di intervalli in cui è contenuta la radice: la scelta dell'intervallo in base al segno della funzione comporta una convergenza globale. É più veloce rispoetto al metodo di bisezione, anche se in generale $[a_i, b_i]_{i\to\infty} \nrightarrow 0$, pertanto il criterio di arresto basato sull'ampiezza dell'intervallo non è applicabile.

4.6 Versione linearizzata

Si può approssimare la funzione con una retta per $(x_0, f(x_0))$, ossia $y = f(x_0) + m(x - x_0)$, ottenendo così una versione linearizzata del problema $f(x) = 0$, ossia:

$$\begin{cases} y = f(x_0) + m(x - x_0) \\ y = 0 \end{cases}$$

da cui si ha $x_1 = x_0 - \dfrac{f(x_0)}{m}$. In generale $x_{i+1} = x_i - \dfrac{f(x_i)}{m_i}$, dove, a seconda della scelta di m_i, si ottengono:

- Il metodo delle corde ($m_i = m$ costante);
- Il metodo delle secanti;
- Il metodo di Newton.

4.7 Il metodo delle secanti

Definisco con m_i il coefficiente angolare della retta passante per $(x_i, f(x_i))$, $(x_{i-1}, f(x_{x-1}))$, ossia:

$$\begin{cases} m_i = \dfrac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \\ x_{i+1} = x_i - \dfrac{f(x_i)}{m_i} = x_i - f(x_i)\dfrac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})} \end{cases}$$

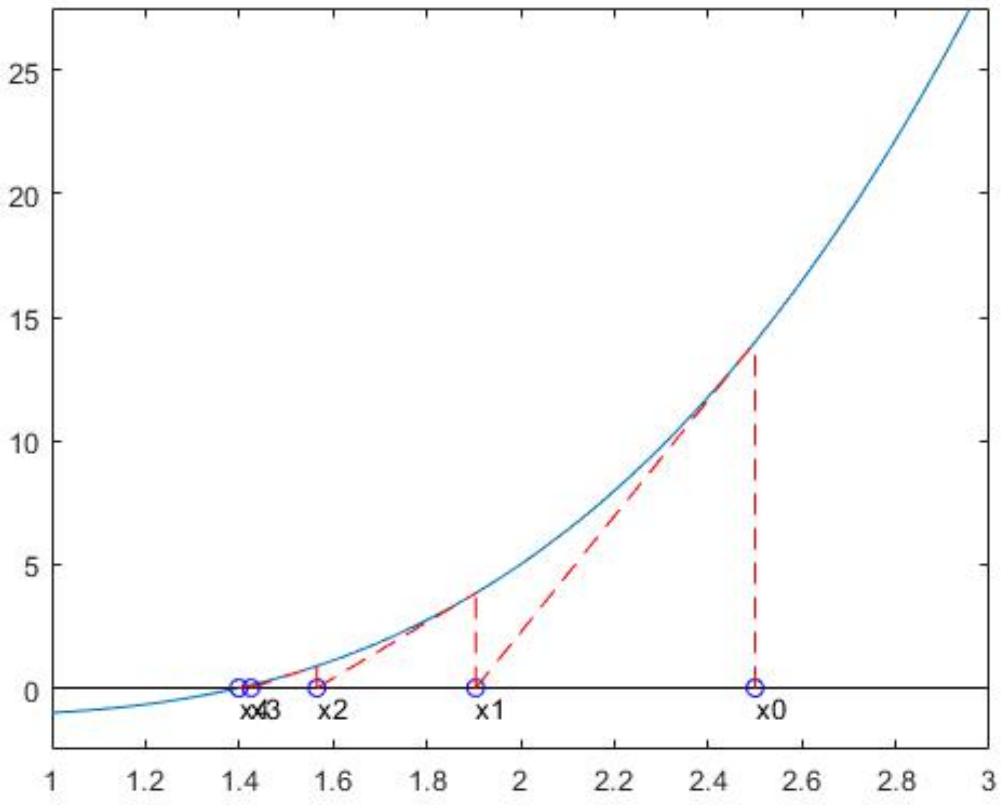
Questo metodo può essere visto come una variante della regula falsi in cui sono richieste due approssimazioni iniziali senza alcun'altra condizione e senza la necessità di controllare il segno di $f(x)$. La convergenza del metodo è garantita se le approssimazioni iniziali sono "abbastanza vicine" alla radice α, pertanto si ha una convergenza locale.

4.8 Metodo di Newton

Definisco con m_i la derivata di f in x_i, ossia:

$$\begin{cases} m_i = f'(x_i) \\ x_{i+1} = x_i - \dfrac{f(x_i)}{f'(x_i)} \end{cases}$$

Geometricamente si prende come nuova approssimazione l'intersezione delle ascisse con la retta tangente ad f in $(x_i, f(x_i))$. Alla i-esima iterazione questo metodo richiede due valutazioni funzionali: $f(x_i)$ e $f'(x_i)$. L'aumento del costo computazionale è compensato dal fatto che la convergenza (locale) è di ordine superiore al primo (in generale è quadratico).



4.9 Metodi di iterazione funzionale

La ricerca degli zeri di una funzione f è ricondotta allo studio dei punti fissi di un'opportuna funzione g, infatti $f(\alpha) = 0 \Leftrightarrow g(\alpha) = \alpha$. La successione delle approssimazioni sarà definita, con x_0 assegnato, come $\forall i \in [1, n], x_i = g(x_{i-1})$. la funzione di iterazione g

non è unica e non può essere costruita nei modi più diversi, ma non tutti daranno luogo a strumenti efficienti. Bisogna studiare sotto quali condizioni la successione delle iterate appartenga sempre al dominio di f e sia convergente ad α. Per esempio:

- Metodo delle corde:

$$f(x) = 0 \Leftrightarrow -\frac{f(x)}{m} = 0 \Leftrightarrow x - \frac{f(x)}{m} = x \Leftrightarrow g(x) = x - \frac{f(x)}{m}$$

- Metodo di Newton:

$$f(x) = 0 \Leftrightarrow -\frac{f(x)}{f'(x)} = 0 \Leftrightarrow x - \frac{f(x)}{f'(x)} = x \Leftrightarrow g(x) = x - \frac{f(x)}{f'(x)}$$

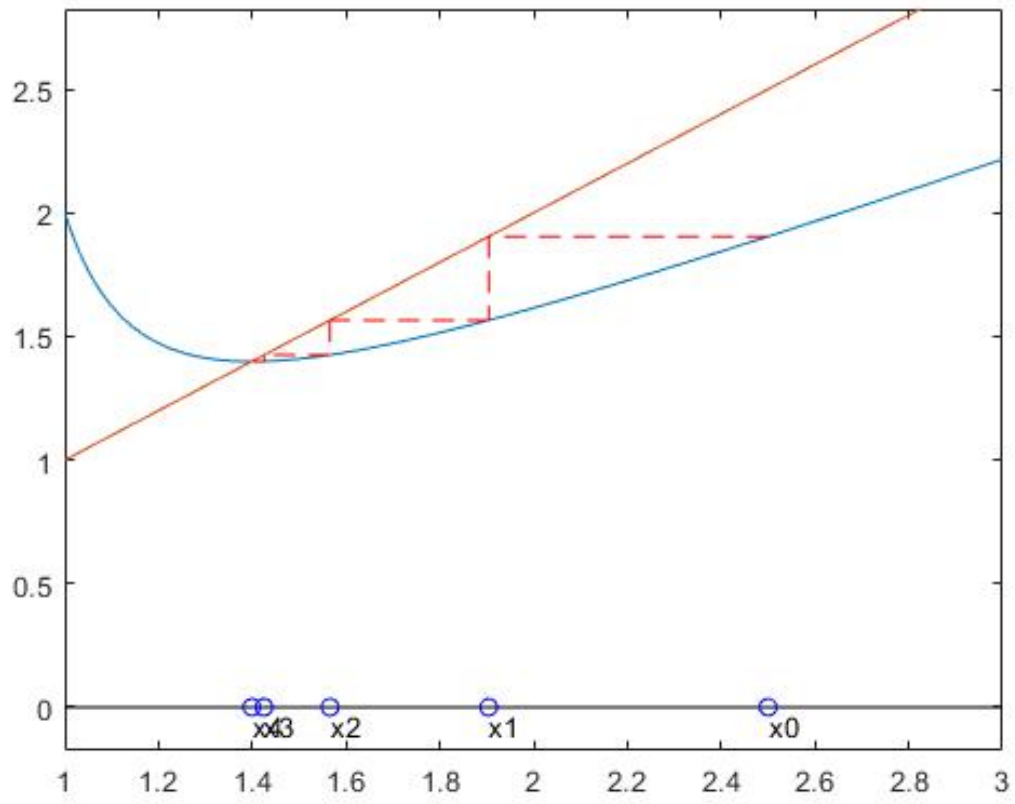


Figura 4.1: Esempio di convergenza monotona con il metodo iterativo di Newton

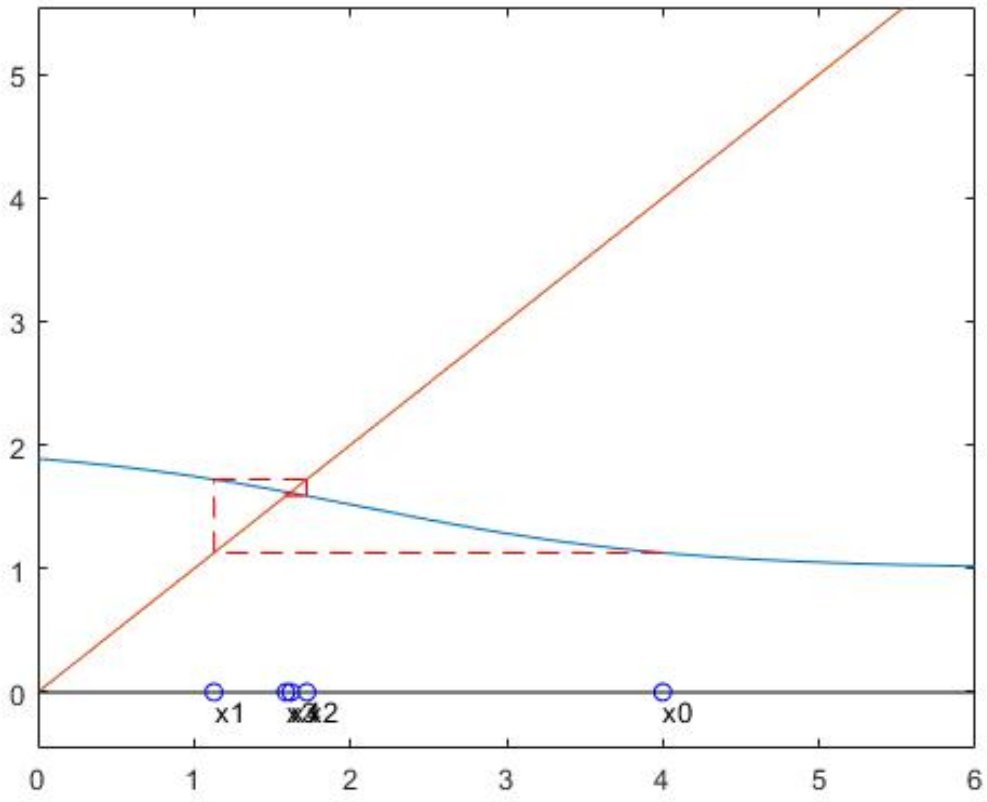


Figura 4.2: Esempio di convergenza alternata con il metodo iterativo di Newton

4.10 Convergenza

Questo che abbiamo trovato è un risultato importante teoricamente, ma nella pratica è difficile stabilire a priori l'intervallo in cui sono soddisfatte le ipotesi. La convergenza può esserci in insiemi molto più grandi di quelli in $|g'(\alpha)|< 1$, pertanto è una

4.10.1 Teorema di Ostrowski

Sia α un punto fisso di $g \in C^1[\alpha-\rho, \alpha+\rho]$. Se $\forall x \in [\alpha-\rho, \alpha+\rho], |g'(x)|< 1$ allora $\forall x_0 \in [\alpha-\rho, \alpha+\rho]$ la successione delle iterate generata da g è tale da:

- $x_i \to \alpha$ unico punto fisso in g;
- $x_i \in [\alpha-\rho, \alpha+\rho]$.

Dimostrazione

Per ipotesi g è una contrazione per cui il punto fisso è unico (si dimostra per assurdo). Per dimostrare che la successione converge si considera

$$x_{i+1} - \alpha = g(x_i) - g(\alpha) = g'(\eta_i)(x_i - \alpha)$$

Dove $\eta_i \in (\alpha, x_i)$. Essendo $x_i \in [\alpha - \rho, \alpha + \rho]$, si ha che $|g'(\eta_i)| < M < 1$, per cui

$$|x_{i+1} - \alpha| < M|x_{i+1} - \alpha| < M^2|x_{i-1} - \alpha| < ... < M^{i+1}|x_0 - \alpha| \Leftarrow \lim_{i \leftarrow \infty} |x_{i+1} - \alpha| = 0$$

Inoltre, dalla continuità di g', si ha $\lim_{i \leftarrow \infty} \frac{|x_{i+1} - \alpha|}{|x_i - \alpha|} = \lim_{i \leftarrow \infty} g'(\eta_i) = g'(\alpha)$

4.11 Condizione di convergenza o divergenza locale

La convergenza può esserci in insiemi molto più grandi di quelli in $|g'(\alpha)| < 1$, pertanto è una condizione sufficiente, avendo così una convergenza locale. Sappiamo che $x_{i+1} - \alpha + g'(\eta_i)(x_i - \alpha)$, dove $\eta_i \in (\alpha, x_i)$. Se $|g'(\alpha)| > 1$, allora $|\alpha - x_{i+1}| > |\alpha - x_i|$, e quindi si avrebbe una divergenza locale.

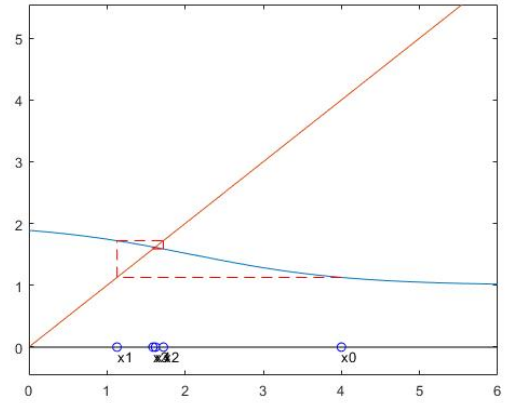


Figura 4.3: Convergenza alternata (ossia $-1 < g'(\alpha) < 0$)

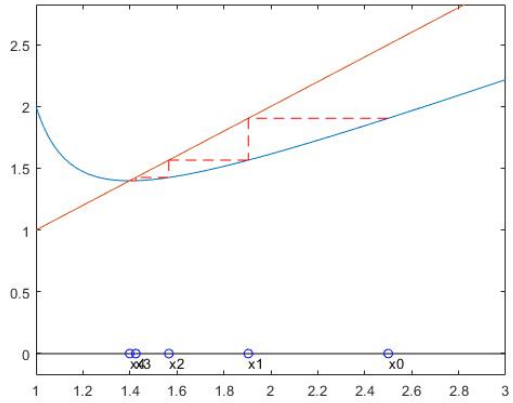


Figura 4.4: Convergenza monotona (ossia $0 < g'(\alpha) < 1$)

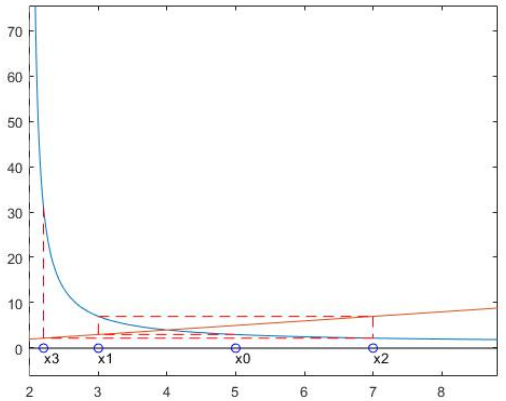


Figura 4.5: Divergenza alternata (ossia $g'(\alpha) < -1$)

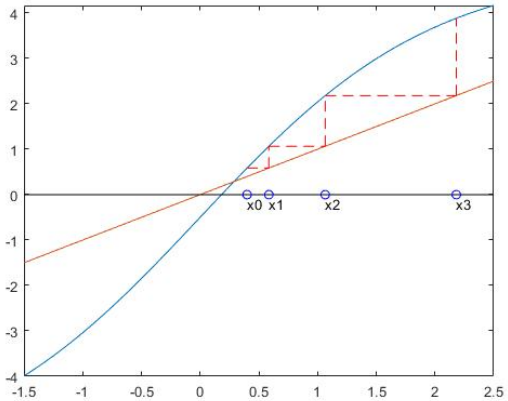


Figura 4.6: Divergenza monotona (ossia $g'(\alpha) > 1$)

4.12 Ordine di convergenza

Per i metodi di iterazione funzionale è possibile anche dare una relazione tra ordine del metodo e molteplicità di α rispetto a g'.

4.12.1 Teorema

Sia $\alpha \in$ I (opportuno intervallo) punto fisso di $g \in \mathrm{C}^p[\mathrm{I}]$ con $p > 2$. Se per un punto $x_0 \in$ I la successione $\{x_0\}$ è convergente e se $g'(\alpha) = g''(\alpha) = ... = g^{(p-1)}(\alpha) = 0$ e $g^{(p)}(\alpha) \neq 0$ allora il metodo ha ordine di convergenza p e risulta $\lim_{i\to\infty} \dfrac{|x_{i+1} - \alpha|}{|x_i - \alpha|} = \dfrac{g^{(p)}(\alpha)}{p!}$

Dimostrazione

Dallo sviluppo di Taylor (tenuto conto che $x_{i+1} = g(x_i)$) si ha in generale che

$$x_{i+1} - \alpha = g(x_i) - g(\alpha) =$$
$$= g'(\alpha)(x_i - \alpha) + g''(\alpha)\frac{(x_i - \alpha)^2}{2!} + ... + g^{(p-1)}(\alpha)\frac{(x_i - \alpha)^{p-1}}{(p-1)!} + g^{(p)}(\eta)\frac{(x_i - \alpha)^p}{p!}$$

Dove $\eta \in (x_i, \alpha)$. Quindi se valgono le ipotesi del teorema di ha la tesi.

Osservazione

A parità di ordine di convergenza p, quanto più piccola risulterà la quantità $\dfrac{g^{(p)}(\alpha)}{p!}$ tanto più veloce sarà la convergenza delle iterate ad α.

4.13 Convergenza del metodo di Newton

Il metodo di Newton può essere visto come un metodo di iterazione funzionale con la funzione g data da $g(x) = x - \dfrac{f(x)}{f'(x)}$. Osservando che se $f \in C^2$ e $f'(\alpha) \neq 0$ (ovvero α è una radice semplice), $g'(x) = \dfrac{f''(x)f(x)}{(f'(x))^2} \Rightarrow g'(\alpha) = 0$, quindi il metodo è sempre localmente convergente e la convergenza è almeno quadratica. Per radici doppie ($f'(\alpha) = 0$), o multiple in generale, la convergenza si riduce a lineare.

4.13.1 Teorema

Sia $f \in C^2[\alpha, \alpha + \rho]$ tale che:

- $f(x)f''(x) > 0$ con $x \in (\alpha, \alpha + \rho]$;
- $f'(x) \neq 0$ con $x \in (\alpha, \alpha + \rho]$.

Allora $\forall x_0 \in (\alpha, \alpha + \rho]$ la successione originata dal metodo di newton decresce monotonicamente ad α. Per gli intorni sinistri $[\alpha - \rho, \alpha)$ si ottiene una successione che converge in modo monotono crescente ad α.

Capitolo 5

Richiami di algebra lineare

5.1 Proprietà delle matrici

Sia $A \in M_{n*n}(\mathbb{R})$, si dice che A è:

- Simmetrica se $A = A^T$;
- Ortogonale se $A^T A = I$;
- (Simmetrica) definita positiva se $\forall x \neq 0, x^T A x > 0$;
- A diagonale dominante in senso forte se $\forall i \in [1,n], |a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|$;
- A diagonale dominante in senso debole se $\forall i \in [1,n], |a_{ii}| \geq \sum_{j=1, j\neq i}^{n} |a_{ij}|$;

Sia $B \in M_{n*n}(\mathbb{C})$, si dice che B è:

- Hermitiana se $B = B^*$, dove $B\hat{}* = (\bar{b}_{ij})_{i,j\in[1,n]}$;
- Unitaria se $B^* B = I$;
- (Simmetrica) definita positiva se $\forall x \neq 0, x^* B x > 0$;
- A diagonale in senso forte se
  $\forall i \in [1,n], |b_{ii}| \geq \sum_{j=1, j\neq i}^{n} |b_{ij}| \wedge \exists k \in [1,n], |b_{ii}| > \sum_{j=1, j\neq i}^{n} |b_{ij}| \wedge$;

5.1.1 Teorema

Sia $A \in M_{n*n}(\mathbb{R})$ simmetrica, allora le seguenti definizioni sono equivalenti:

- A è detta positiva,
- Gli autovalori di A sono reali e positivi;
- (Criterio di Sylvester) i determinanti dei minori principali di A verificano che $\forall k \in [1,n], \det(A_k) > 0$.

5.2 Norma di vettore

Sia X uno spazio lineare e $f : X \to \mathbb{R}$ una funzionale tale che:

- $f(x) = 0 \Leftrightarrow x = 0$;
- $\forall x \in \mathrm{X}, f(x) \geq 0$;
- $\forall x, y \in \mathrm{X}, f(x + y) \leq f(x) + f(y)$;
- $\forall \alpha \in \mathbb{R}, \forall x \in \mathrm{X}, f(\alpha X) = |\alpha| f(x)$ (nel caso in cui lavorassimo nei complessi, $\forall z \in \mathbb{C}, \forall x \in X, f(zx) = |z| f(x)$).

Allora X è detto spazio lineare normato (SLN) e f è detta norma. In particolare, per $\mathrm{X} = \mathbb{R}^n$, si usano le norme p o Hölderiane: $||x||_p := (|x_1|^p + |x_2|^p + ... + |x_n|^p)^{\frac{1}{p}}$. In particolare:

- Per $p = 1, ||x||_1 = |x_1| + |x_2| + ... + |x_n|$;
- Per $p = 2, ||x||_2 = \sqrt{(x_1)^2 + (x_2)^2 + ... + (x_n)^2}$;
- Per $p = \infty, ||x||_\infty = \max\{|x_1|, |x_2|, ..., |x_n|\}$;

5.3 Norme equivalenti

Siano f e g due norme di $\mathbb{R}^n$. Se esistono $c, \mathrm{C} \in \mathbb{R}^+$ tali che $\forall x \in \mathbb{R}^n, cg(x) \leq f(x) \leq \mathrm{C}g(x)$, allora f e g si dicono norme equivalenti. Da qui abbiamo che $||\cdot||_1, ||\cdot||_2, ||\cdot||_\infty$ sono norme equivalenti ma con diversa geometria:

- $\forall x \in \mathbb{R}^n, ||x||_\infty \leq ||x||_1 \leq n||x||_\infty$;
- $\forall x \in \mathbb{R}^n, ||x||_\infty \leq ||x||_2 \leq \sqrt{n}||x||_\infty$;
- $\forall x \in \mathbb{R}^n, \frac{||x||_1}{\sqrt{n}} \leq ||x||_2 \leq n||x||_1$.

5.4 Norma di matrici compatibile o consistente

Sia $||\cdot||_{\mathbb{R}^n}$ una norma vettoriale, $||\cdot||_{\mathbb{R}^{n*n}}$ si dice compatibile o consistente con $||\cdot||_{\mathbb{R}^n}$ se $\forall \mathrm{A}, \mathrm{B} \in \mathrm{M}_{n*n}(\mathbb{R})$ e $\forall x \in \mathbb{R}^n$ si ha:

- $||\mathrm{A}x||_{\mathbb{R}^n} \leq ||\mathrm{A}||_{\mathbb{R}^{n*n}} ||x||_{\mathbb{R}^n}$;
- $||\mathrm{AB}||_{\mathbb{R}^{n*n}} \leq ||\mathrm{A}||_{\mathbb{R}^{n*n}} ||\mathrm{B}||_{\mathbb{R}^{n*n}}$.

5.5 Norma di matrici indotta

Sia $||\cdot||_{\mathbb{R}^{n*n}}$ compatibile, si dice indotta da $||\cdot||_{\mathbb{R}^n}$ se
$\forall \mathrm{A} \in \mathrm{M}_{n*n}(\mathbb{R}), \exists x \in \mathbb{R}^n : ||\mathrm{A}x||_{\mathbb{R}^n} = ||\mathrm{A}||_{\mathbb{R}^{n*n}} ||x||_{\mathbb{R}^n}$.

5.6 Norma di matrici naturale

Sia data $||\cdot||_{\mathbb{R}^n}$ una norma vettoriale, il funzionale $||\cdot||: \mathbb{R}^{n*n} \to \mathbb{R}$ tale che $||\mathrm{A}||= \sup_{x\neq 0} \frac{||Ax||_{\mathbb{R}^n}}{||x||_{\mathbb{R}^n}=1} = \max_{||x||_{\mathbb{R}^n}} ||Ax||_{\mathbb{R}^n}$ è una norma, detta norma naturale. Da notare che ogni norma naturale è indotta, infatti, sia $y : ||\mathrm{A}\bar{y}||= \max_{||x||=1} ||\mathrm{A}|| \Rightarrow ||\mathrm{A}\bar{y}||= ||\mathrm{A}||= ||\mathrm{A}||||\bar{y}||$

5.7 Principali norme indotte naturali

Sia $\mathrm{A} \in \mathrm{M}_{n*n}(\mathbb{R})$, dove $\rho(\mathrm{A}) = \max_{k\in[1,n]} |\lambda_k(\mathrm{A})|$ è il suo raggio spettrale, abbiamo che:

- $||\cdot||_1$ induce $||\mathrm{A}||_1= \max_{j\in[1,n]} \sum_{i=1}^{n} |a_{ij}|$;
- $||\cdot||_\infty$ induce $||\mathrm{A}||_\infty= \max_{i\in[1,n]} \sum_{j=1}^{n} |a_{ij}|$;
- $||\cdot||_2$ induce $||\mathrm{A}||_2= \sqrt{\rho(\mathrm{A}^{\mathrm{T}}\mathrm{A})}$.

5.8 Proprieà delle norme naturali

- Sia $\mathrm{A} \in \mathrm{M}_{n*n}(\mathbb{R}), \rho(\mathrm{A}) := \inf\{||\mathrm{A}||: ||\cdot|| \text{ norma naturale}\}$, ovvero $\forall \epsilon > 0, \exists ||\cdot|| \text{ norma naturale} : \rho(\mathrm{A}) \leq ||\mathrm{A}|| \leq \rho(\mathrm{A}) + \epsilon$, da cui segue che $\forall \mathrm{A} \in \mathrm{M}_{n*n}(\mathbb{R}), \forall ||\cdot|| \text{ norma indotta}, \rho(\mathrm{A}) \leq ||\mathrm{A}||$;
- Sappiamo che una matrice $\mathrm{A} \in \mathrm{M}_{n*n}(\mathbb{R})$ si dice convergente se $\lim_{k\to\infty} \mathrm{A}^k = 0$, da cui abbiamo che una matrice è convergente se e solo se $\rho(\mathrm{A}) \leq 1$;
- Proprietà di invarianza della norma rispetto alle trasformazioni ortogonali:
  - Sia $x \in \mathbb{R}^n$, sia $\mathrm{U} \in \mathrm{M}_{n*n}(\mathbb{R})$ ortogonale, abbiamo che $||x||_2= ||\mathrm{U}x||_2$;
  - Sia $\mathrm{A} \in \mathrm{M}_{n*n}(\mathbb{R})$, siano $\mathrm{U}, \mathrm{V} \in \mathrm{M}_{n*n}(\mathbb{R})$ ortogonali, abbiamo che $||\mathrm{A}||_2= ||\mathrm{UAV}||_2$.

Capitolo 6

Cerchi di Gerschgorin e autovalori

6.1 Primo teorema di Gerschgorin

Sia $A \in M_{n*n}(\mathbb{R})$, i suoi autovalori λ sono contenuti nell'unione $\bigcup_{i=1}^{n} K_i$, dove

$$\forall i \in [1,n], K_i := \{z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{j=1, j\neq i}^{n} |a_{ij}|\}$$

È detto cerchio di Gerschgorin con centro a_{ii} e raggio $\sum_{j=1, j\neq i}^{n} |a_{ij}|$. Inoltre sono contenuti nell'unione di $\bigcup_{i=1}^{n} H_i$, dove

$$\forall i \in [1,n], H_i := \{z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{j=1, j\neq i}^{n} |a_{ji}|\}$$

Da cui abbiamo che $\lambda \in (\bigcup_{i=1}^{n} K_i) \cap (\bigcup_{i=1}^{n} H_i)$.

6.2 Secondo teorema di Gerschgorin

Sia M_1 l'unione di k cerchi e M_2 l'unione di $n-k$ cerchi, se $M_1 \cap M_2 = \emptyset$, allora k autovalori sono in M_1 e $n-k$ autovalori sono in M_2.

6.3 Terzo teorema di Gerschgorin

Sia $A \in M_{n*n}(\mathbb{R})$ irriducibile (non è scomponibile), se λ si trova nella frontiera dell'unione dei cerchi, allora ogni λ_i si trova nella frontiera di uno dei cerchi.

6.4 Proprietà

- Sia $A \in M_{n*n}(\mathbb{R})$ a diagonale dominante in senso forte, allora A è non singolare, ossia $\det(A) \neq 0 \Leftrightarrow \lambda_i \neq 0$;

- Sia $A \in M_{n*n}(\mathbb{R})$ tale che:
  - $A = A^T$;
  - A è a diagonale dominante in senso forte;
  - $\forall i \in [1, n], a_{ii} > 0$.

  Allora A è definita positiva, ossia $\forall x \in \mathbb{R}, x \neq 0, x^T Ax > 0 \Leftrightarrow \lambda_i > 0 \Leftrightarrow \det(A) > 0$.

Capitolo 7

Sistemi lineari quadrati

Sia $\mathrm{A} \in \mathrm{M}_{n*n}(\mathbb{R})$, sia $b \in \mathbb{R}^n$, se $\det(\mathrm{A}) \neq 0$ allora $\exists!\, x \in \mathbb{R}^n : \mathrm{A}x = b$. Molti modelli matematici significativi sono di tipo lineare. I sistemi lineari nascono da contesti diversi, e pertanto è importante disporre di un'ampia varietà di algoritmi in modo da scegliere il più adatto al problema specifico (stabilità, occupazione di memoria, velocità). I metodi per i sistemi lineari si dividono in due gruppi:

- Metodi diretti, che si basano sull'idea di trasformare il sistema attraverso un numero finito di operazioni in un sistema equivalente di cui sia esplicitamente calcolabile la soluzione (in assenza di errori arrotondamento forniscono la soluzione esatta);
- Metodi iterativi, la cui soluzione è ottenuta come limite di una successione (permettono di sfruttare la sparsità della matrice in quanto, al contrario dei metodi diretti, tale matrice non viene modificata).

7.1 Condizionameto

Sia $\mathrm{A} \in \mathrm{M}_{n*n}(\mathbb{R})$ tale che A è non singolare e $\det(\mathrm{A}) \neq 0$, e sia $b \in \mathbb{R}^n$ tale che $b \neq 0$, allora esiste un $x \in \mathbb{R}^n$, con $x \neq 0$, tale che $\mathrm{A}x = b$. Poniamo di calcolare $(\mathrm{A} + \delta\mathrm{A})(x + \delta x) = b + \delta b$ con $\delta\mathrm{A}$ tale che $||\delta\mathrm{A}||$ è "piccolo" e $\delta\mathrm{A}$ non è singolare, e con δb tale che $||\delta b||$ è "piccolo"

$$
\begin{aligned}
&(\mathrm{A} + \delta\mathrm{A})(x + \delta x) = b + \delta b \\
&\mathrm{A}x + \mathrm{A}\delta x + \delta\mathrm{A}x + \delta\mathrm{A}\delta x = b + \delta b \\
&\mathrm{A}\delta x + \delta\mathrm{A}x + \delta\mathrm{A}\delta x = \delta b \\
&(\mathrm{A} + \delta\mathrm{A})\delta x = \delta b - \delta\mathrm{A}x
\end{aligned}
$$

Sia $\dfrac{||\delta x||}{||x||}$ la perturbazione dei risultati, allora

$$
\begin{aligned}
&||\delta x|| \leq ||(\mathrm{A} + \delta\mathrm{A})^{-1}||(||\delta b|| + ||\delta\mathrm{A}||||x||) \leq \\
&\leq ||(\mathrm{I} + \mathrm{A}^{-1}\delta\mathrm{A})^{-1}||||\mathrm{A}||(\frac{||\delta b||}{||b||}||\mathrm{A}||||x|| + \frac{||\delta\mathrm{A}||}{||\mathrm{A}||}||x||||\mathrm{A}||) \leq \\
&\leq ||(\mathrm{I} + \mathrm{A}^{-1}\delta\mathrm{A})^{-1}||||\mathrm{A}^{-1}||||\mathrm{A}||(\frac{||\delta b||}{||b||}||x|| + \frac{||\delta\mathrm{A}||}{||\mathrm{A}||}||x||) \Rightarrow \\
&\Rightarrow \frac{||\delta x||}{||x||} \leq ||(\mathrm{I} + \mathrm{A}^{-1}\delta\mathrm{A})^{-1}||||\mathrm{A}^{-1}||||\mathrm{A}||(\frac{||\delta b||}{||b||} + \frac{||\delta\mathrm{A}||}{||\mathrm{A}||})
\end{aligned}
$$

Sapendo che $||(\mathrm{I}+\mathrm{A}^{-1}\delta\mathrm{A})^{-1}||\leq \dfrac{1}{1-||\mathrm{A}^{-1}||||\delta\mathrm{A}||}(\cong 1)$ e ponendo
$\mu(\mathrm{A}) = ||\mathrm{A}^{-1}||||\mathrm{A}||$, dove $\mu(\mathrm{A})$ è detto numero di condizionamento, abbiamo

$$\frac{||\delta x||}{||x||} \leq \mu(\mathrm{A})(\frac{||\delta b||}{||b||} + \frac{||\delta\mathrm{A}||}{||\mathrm{A}||})\frac{1}{1-||\mathrm{A}^{-1}||||\delta\mathrm{A}||}$$

Da notare che è possibile usare un qualsiasi norma indotta ($||\cdot||_1, ||\cdot||_2, ||\cdot||_\infty$). Si noti che $\mu(\mathrm{A}) \geq ||\mathrm{A}^{-1}||||\mathrm{A}||\geq ||\mathrm{A}^{-1}\mathrm{A}||= ||\mathrm{I}||= 1 \Rightarrow \mu(\mathrm{A}) \geq 1$

7.1.1 Teorema

Sia $\mathrm{A} \in \mathrm{M}_{n*n}(\mathbb{R})$, sia $||\cdot||$ norma indotta tale che $||\mathrm{A}||< 1$, allora:

- $\mathrm{I}+\mathrm{A}$ è invertibile;
- $\dfrac{1}{1+||\mathrm{A}||} \leq ||(\mathrm{I}+\mathrm{A})^{-1}||\leq \dfrac{1}{1-||\mathrm{A}||}$.

7.2 Metodo di Cramer

Il metodo di Cramer è un metodo per calcolare la soluzione del sistema $\mathrm{A}x+b$. Questo metodo è inefficiente (richiede $\mathrm{O}(n!)$ prodotti), anche se in qualche caso il calcolo è semplice: in particolare, si può usare in sistemi con matrici triangolari superiori ed inferiori.

7.3 Algoritmo di sostituzione in avanti

Sia $\mathrm{L}x = b$ con $\mathrm{L} \in \mathrm{M}_{n*n}(\mathbb{R})$ triangolare inferiore non singolare

$$\begin{cases} l_{11}x_1 = b_1 \\ l_{21}x_1 + l_{22}x_2 = b_2 \\ \vdots \\ l_{i1}x_1 + l_{i2}x_2 + ... + l_{ii}x_i = b_i \\ \vdots \\ l_{n1}x_1 + l_{n2}x_2 + ... + l_{nn}x_n = b_n \end{cases} \rightarrow \begin{cases} x_1 = \dfrac{b_1}{l_{11}} \\ x_2 = \dfrac{b_2 - l_{21}x_1}{l_{22}} \\ \vdots \\ x_i = \dfrac{b_i - (l_{i1}x_1 + l_{i2}x_2 + ... + l_{i,i-1}x_{i-1})}{l_{ii}} \\ \vdots \\ x_n = \dfrac{b_n - \displaystyle\sum_{j=1}^{n-1} l_{nj}x_j}{l_{nn}} \end{cases}$$

Costo computazionale

Si calcola il numero di prodotti/divisioni eseguite in funzione di n e si assume come costo il termine con la potenza più alta:

$$\begin{cases} \displaystyle\sum_{i=2}^{n}(i-1) = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} \text{ prodotti} \\ n \text{ divisioni} \end{cases} \Rightarrow \frac{n^2}{2} \text{ operazioni}$$

Algoritmo di sostituzione in avanti

```
1: function ALGORITMOAVANTI(L, b)
2:     for i = 1 to n do
3:         x_i ← b_i
4:         for j = 1 to i − 1 do
5:             x_i ← x_i − l_ij x_j
6:         end for
7:         x_i ← x_i / l_ii
8:     end for
9:     return x
10: end function
```

7.4 Algoritmo di sostituzione all'indietro

Sia $\mathrm{R}x = b$ con $\mathrm{R} \in \mathrm{M}_{n*n}(\mathbb{R})$ triangolare superiore non singolare

$$\begin{cases} r_{11}x_1 + r_{12}x_2 + ... + r_{1n}x_n = b_1 \\ \vdots \\ r_{ii}x_i + r_{i,i+1}x_{i+1} + ... + r_{in}x_n = b_i \\ \vdots \\ r_{n-1,n-1}x_{n-1} + r_{n-1,n}x_n = b_{n-1} \\ r_{nn}x_n = b_n \end{cases} \rightarrow \begin{cases} x_n = \dfrac{b_n}{r_{nn}} \\ x_{n-1} = \dfrac{b_{n-1} - r_{n-1,n}x_n}{r_{n-1,n-1}} \\ \vdots \\ x_i = \dfrac{b_i - (r_{i,i+1}x_{i+1} + ... + r_{in}x_n)}{r_{ii}} \\ \vdots \\ x_1 = \dfrac{b_1 - \sum_{j=2}^{n} r_{1j}x_j}{r_{11}} \end{cases}$$

Algoritmo di sostituzione all'indietro

```
1: function ALGORITMOAVANTI(L, b)
2:     for i = n down to 1 do
3:         x_i ← b_i
4:         for j = i + 1 to n do
5:             x_i ← x_i − r_ij x_j
6:         end for
7:         x_i ← x_i / r_ii
8:     end for
9:     return x
10: end function
```

Costo computazionale

Si calcola il numero di prodotti/divisioni eseguite in funzione di n e si assume come costo il termine con la potenza più alta:

$$\begin{cases} \sum_{i=2}^{n}(i-1) = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} \text{ prodotti} \\ n \text{ divisioni} \end{cases} \quad \Rightarrow \frac{n^2}{2} \text{ operazioni}$$

7.5 Fattorizzazione LR

L'idea di base è quella di trasformare un problema complesso in uno equivalente più semplice. Sia $\mathrm{A} \in \mathrm{M}_{n*n}(\mathbb{R})$, se esistono $\mathrm{L} \in \mathrm{M}_{n*n}(\mathbb{R})$ triangolare inferiore ed $\mathrm{R} \in \mathrm{M}_{n*n}(\mathbb{R})$ triangolare superiore tale che $\mathrm{A} = \mathrm{LR}$ (fattorizzazione LR), allora $\mathrm{A}x = b \Leftrightarrow \mathrm{LR}x = b \Leftrightarrow$ $\begin{cases} \mathrm{L}y = b \\ \mathrm{R}x = y \end{cases}$

$$\underbrace{\begin{bmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{bmatrix}}_{\mathrm{L}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ 0 & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & r_{nn} \end{bmatrix}}_{\mathrm{R}} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}}_{\mathrm{A}}$$

Vengono usati due metodi per fissare n incognite di partenza:

- $\forall i \in [1, n], r_{ii} = 1$ (fattorizzazione alla Crout);
- $\forall i \in [1, n], l_{ii} = 1$ (fattorizzazione alla Dolittle);

Costo computazionale

Richiede $\frac{n^3}{3}$ operazioni, quindi è instabile e richiede strategie di pivoting.

7.6 Teorema di fattorizzazione

Sia $\mathrm{A} \in \mathrm{M}_{n*n}(\mathbb{R})$, se $\forall k \in [1, n-1], \det(\mathrm{A}_k) \neq 0$ (A_k minore principale di ordine k), allora esiste ed è unica la fattorizzazione alla Dolittle e $\det(\mathrm{A}) = \prod_{i=1}^{n} r_{ii}$ (le ipotesi del teorema non sono di facile verifica, ma sono soddisfatte dalle matrici simmetriche e definite positive e dalle matrici a diagonale dominante in senso debole e non singolari).

7.6.1 Teorema

Sia $\mathrm{A} \in \mathrm{M}_{n*n}(\mathbb{R})$ simmetrica e definita positiva, allora $\forall k \in [1, n-1], \det(\mathrm{A}_k) > 0$.

7.7 Fattorizzazione LR per matrici simmetriche

Sia $\mathrm{A} \in \mathrm{M}_{n*n}(\mathbb{R})$ simmetrica ($\mathrm{A} = \mathrm{A}^{\mathrm{T}}$), definiamo

$$\mathrm{D} := \begin{bmatrix} r_{11} & 0 & 0 & \dots & 0 \\ 0 & r_{22} & 0 & \dots & 0 \\ 0 & 0 & r_{33} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & r_{nn} \end{bmatrix}; \mathrm{U} := \begin{bmatrix} 1 & \frac{r_{12}}{r_{11}} & \frac{r_{13}}{r_{11}} & \dots & \frac{r_{1n}}{r_{11}} \\ 0 & 1 & \frac{r_{23}}{r_{22}} & \dots & \frac{r_{2n}}{r_{22}} \\ 0 & 0 & 1 & \dots & \frac{r_{3n}}{r_{33}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

$\mathrm{R} = \mathrm{DU}$, $\mathrm{A} = \mathrm{LDU}$ e $\mathrm{A}^{\mathrm{T}} = \mathrm{U}^{\mathrm{T}}\mathrm{DL}^{\mathrm{T}}$, ma $\mathrm{A} = \mathrm{A}^{\mathrm{T}} \Rightarrow \mathrm{L} = \mathrm{U}^{\mathrm{T}}$

Il calcolo di L risulta semplificato, e la fattorizzazione richiede $\frac{n^3}{6}$ operazioni.

7.8 Fattorizzazione di Choleski

Sia $\mathrm{A} \in \mathrm{M}_{n*n}(\mathbb{R})$ simmetrica e definita positiva, ossia $\mathrm{A} = \mathrm{A}^{\mathrm{T}}$ e $\forall x \in \mathbb{R}^n, x \neq 0, x^{\mathrm{T}}\mathrm{A}x > 0$, dal teorema di fattorizzazione $\det(\mathrm{A}) = \prod_{i=1}^{n} r_{ii}$, abbiamo che

$\forall i \in [1, n], r_{ii} > 0$; da cui si pone

$$\mathrm{D}^{\frac{1}{2}} := \begin{bmatrix} \sqrt{r_{11}} & 0 & \dots & 0 \\ 0 & \sqrt{r_{22}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sqrt{r_{nn}} \end{bmatrix} = \mathrm{diag}(\sqrt{r_{11}}, \sqrt{r_{22}}, ..., \sqrt{r_{nn}})$$

$\mathrm{A} = \mathrm{LDU} = \mathrm{LDL}^{\mathrm{T}} = \mathrm{LD}^{\frac{1}{2}}\mathrm{D}^{\frac{1}{2}}\mathrm{L}^{\mathrm{T}}$, e posto $\mathrm{S} = \mathrm{LD}^{\frac{1}{2}}$, abbiamo $\mathrm{A} = \mathrm{SS}^{\mathrm{T}}$. Da notare che se, date $\mathrm{M}, \mathrm{N} \in \mathrm{M}_{n*n}(\mathbb{R})$, $\mathrm{M} = \mathrm{NN}^{\mathrm{T}}$ e $\det(\mathrm{N}) \neq 0$, allora M è simmetrica e definita positiva, ossia $\forall x \in \mathbb{R}^n, x \neq 0, x^{\mathrm{T}}\mathrm{M}x = x^{\mathrm{T}}\mathrm{NN}^{\mathrm{T}}x = ||\mathrm{N}^{\mathrm{T}}x||_2^2 > 0$

$$\underbrace{\begin{bmatrix} s_{11} & 0 & ... & 0 \\ s_{21} & s_{22} & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & ... & s_{nn} \end{bmatrix}}_{\mathrm{S}} \underbrace{\begin{bmatrix} s_{11} & s_{12} & ... & s_{1n} \\ 0 & s_{22} & ... & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & s_{nn} \end{bmatrix}}_{\mathrm{S}^{\mathrm{T}}} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & ... & a_{nn} \end{bmatrix}}_{\mathrm{A}}$$

$$a_{ii} = \sum_{k=1}^{i} s_{ik}^2 \Rightarrow s_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} s_{ik}^2}$$

$$\forall i > j, a_{ij} = \sum_{k=1}^{j} s_{ik}s_{kj} \Rightarrow s_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} s_{ik}s_{kj}}{s_{jj}}$$

Costo computazionale

Richiede $\frac{n^3}{6}$ operazioni.

Fattorizzazione di Choleski

```
1: function algoritmoCholeski(A)
2:     for j = 1 to n do
3:         s_jj = a_jj
4:         for k = 1 to j − 1 do
5:             s_jj ← s_jj − s_jk^2
6:         end for
7:         s_jj ← √s_jj
8:         for i = j + 1 to n do
9:             s_ij ← a_ij
10:            for k = 1 to j − 1 do
11:                s_jj ← s_jj − s_ik s_kj
12:            end for
13:            s_ij ← s_ij / s_jj
14:        end for
15:    end for
16:    return S
17: end function
```

Osservazioni

- L'algoritmo di Choleski è stabile;
- Occorre verificare se $a_{ii} - \sum_{k=1}^{i-1} s_{ik}^2 > 0$ (ossia si effettua il controllo $a_{ii} - \sum_{k=1}^{i-1} s_{ik}^2 > toll$) per evitare che si divida per 0 (ossia per $s_{jj} \cong 0$);
- L'algoritmo di Choleski è usato per verificare se la matrice A simmetrica è definita positiva.

7.9 Metodo di eliminazione di Gauss

Come si può passare da A$x = b$ a LR$x = b$? Il metodo di Gauss formalizza l'idea di risolvere il sistema eliminando via via le incognite delle equazioni e trasforma il sistema assegnato in uno equivalente del tipo R$x = y$.

$$\begin{cases} 1x + 4y + 7z = 1 \\ 2x + 5y + 8z = 1 \\ 3x + 6y + 11z = 1 \end{cases} \rightarrow \begin{cases} 1x + 4y + 7z = 1 \\ -3y - 6z = -1 \\ -6y - 10z = -2 \end{cases} \rightarrow \begin{cases} 1x + 4y + 7z = 1 \\ -3y - 6z = -1 \\ 2z = 0 \end{cases}$$

In termini matriciali

$$\underbrace{\left[\begin{array}{ccc|c} 1 & 4 & 7 & 1 \\ 2 & 5 & 8 & 1 \\ 3 & 6 & 11 & 1 \end{array}\right]}_{[\mathrm{A}^{(0)}|b^{(0)}]} \rightarrow \underbrace{\left[\begin{array}{ccc|c} 1 & 4 & 7 & 1 \\ 0 & -3 & -6 & -1 \\ 0 & -6 & -10 & -2 \end{array}\right]}_{[\mathrm{A}^{(1)}|b^{(1)}]} \rightarrow \underbrace{\left[\begin{array}{ccc|c} 1 & 4 & 7 & 1 \\ 0 & -3 & -6 & -1 \\ 0 & 0 & 2 & 0 \end{array}\right]}_{[\mathrm{A}^{(2)}|b^{(2)}]}$$

Anche le trasformazioni possono essere descritte in termini matriciali

$$\mathrm{A}^{(0)} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 11 \end{bmatrix} \rightarrow \mathrm{M}^{(0)} \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{bmatrix}, \mathrm{M}^{(0)}\mathrm{A}^{(0)} = \mathrm{A}^{(1)} = \begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & -6 & -10 \end{bmatrix}$$

$$\mathrm{A}^{(1)} = \begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & -6 & -10 \end{bmatrix} \rightarrow \mathrm{M}^{(0)} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix}, \mathrm{M}^{(1)}\mathrm{A}^{(1)} = \mathrm{A}^{(2)} = \begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & 0 & 2 \end{bmatrix}$$

In generale, dato $\mathrm{A}x = b$, si vogliono costruire $\mathrm{M}_1, \mathrm{M}_2, ..., \mathrm{M}_{n-2}, \mathrm{M}_{n-1}$ tali che

$$\begin{aligned} &\mathrm{M}_{n-1}\mathrm{M}_{n-2}...\mathrm{M}_2\mathrm{M}_1\mathrm{A} = \mathrm{R} \Leftrightarrow \\ &\Leftrightarrow \mathrm{M}_{n-1}\mathrm{M}_{n-2}...\mathrm{M}_2\mathrm{M}_1\mathrm{A}x = \mathrm{M}_{n-1}\mathrm{M}_{n-2}...\mathrm{M}_2\mathrm{M}_1 b \Leftrightarrow \\ &\Leftrightarrow \mathrm{R}x = \mathrm{M}_{n-1}\mathrm{M}_{n-2}...\mathrm{M}_2\mathrm{M}_1 b \end{aligned}$$

In termini algoritmici

$$\begin{cases} [\mathrm{A}^{(0)}|b^{(0)}] = [\mathrm{A}|b] \\ [\mathrm{A}^{(k)}|b^{(k)}] = \mathrm{M}_k[\mathrm{A}^{(k-1)}|b^{(k-1)}], \forall k \in [1, n-1] \end{cases}$$

Dove

$$\forall k \in [1, n-1], \mathrm{M}_k := \mathrm{I} - \alpha^{(k)} e_k^{\mathrm{T}}$$

$$\forall i \in [1, n], \alpha_i^{(k)} := \begin{cases} 0 \text{ se } i \leq k \\ \dfrac{\alpha_{ik}^{(k-1)}}{\alpha_{kk}^{(k-1)}} \text{ se } i > k \end{cases}$$

$$\forall i \in [1, n], (e_k)_i := \begin{cases} 0 \text{ se } i \neq k \\ 1 \text{ se } i = k \end{cases}$$

$\alpha_{kk}^{(k-1)}$ è l'elemento di pivot, ed è cruciale per lo svolgimento del metodo, infatti:

- Il metodo di Gauss si interrompe se l'elemento pivot $\alpha_{kk}^{(k-1)} = 0$;
- Le ipotesi di fattorizzazione ($\forall k \in [1, n-1], \det(\mathrm{A}_k) \neq 0$) garantiscono che $\alpha_{kk}^{(k-1)}$.

Da notare che il metodo di Gauss dà la fattorizzazione LR di A, infatti

$$\begin{aligned} &\mathrm{M}_{n-1}\mathrm{M}_{n-2}...\mathrm{M}_2\mathrm{M}_1\mathrm{A} = \mathrm{R} \rightarrow \\ &\rightarrow \mathrm{A} = (\mathrm{M}_{n-1}\mathrm{M}_{n-2}...\mathrm{M}_2\mathrm{M}_1)^{-1}\mathrm{R} \rightarrow \\ &\rightarrow \mathrm{A} = \underbrace{\mathrm{M}_1^{-1}\mathrm{M}_2^{-1}...\mathrm{M}_{n-2}^{-1}\mathrm{M}_{n-1}^{-1}}_{=\mathrm{L}}\mathrm{R} \rightarrow \\ &\rightarrow \mathrm{A} = \mathrm{LR} \end{aligned}$$

Quindi

$$\mathrm{A}x = b \Leftrightarrow \mathrm{LR}x = b \Leftrightarrow \begin{cases} \mathrm{L}y = b \\ \mathrm{R}x = y \end{cases} \Leftrightarrow \mathrm{R}x = \mathrm{L}^{-1}b \Leftrightarrow$$

$$\Leftrightarrow \underbrace{\mathrm{M}_{n-1}\mathrm{M}_{n-2}...\mathrm{M}_2\mathrm{M}_1}_{\mathrm{R}}\mathrm{A}x = \underbrace{\mathrm{M}_{n-1}\mathrm{M}_{n-2}...\mathrm{M}_2\mathrm{M}_1}_{\mathrm{L}^{-1}} b$$

Metodo di eliminazione di Gauss

```
1: function algoritmoGauss(A)
2:     k ← 1
3:     while k < n and a_kk ≠ 0 do
4:         for i = k + 1 to n do
5:             η ← a_ik / a_kk
6:             a_ik ← η
7:             for i = k + 1 to n do
8:                 a_ij ← a_ij − η a_kj
9:             end for
10:        end for
11:        k ← k + 1
12:    end while
13: end function
```

Costo computazionale

Richiede $\sum_{k=1}^{n-1}(n-k)^2 + \sum_{k=1}^{n-1}(n-k) = \frac{(n-1)n(2n-1)}{6} + \frac{(n-1)n}{2} \cong \frac{n^3}{3}$ operazioni.

Osservazioni

Non si costruiscono esplicitamente le matrici di trasformazione M_k.

7.10 Metodo di Gauss con pivoting e scaling

Sappiamo che, se $\alpha_{kk}^{(k-1)} = 0$, l'algoritmo di Gauss si ferma, anche in casi in cui esiste una soluzione. Un'dea che si può utilizzare è quella di cambiare di posto due righe: la k-esima riga con una delle righe successive per quale $\alpha_{ik}^{(k-1)} \neq 0, i \in [k+1, n]$ (da notare che se $\det(\mathrm{A})) \neq 0$ e $\alpha_{kk}^{(k-1)} \neq 0$, necessariamente qualche elemento $\alpha_{ik}^{(k-1)} \neq 0, i \in [k+1, n]$). Se $\alpha_{kk}^{(k-1)} \neq 0$ ma $|\alpha_{kk}^{(k-1)}|$ è molto piccolo rispetto agli altri elementi, allora l'algoritmo di Gauss può essere instabile. Per estendere il metodo di Gauss a matrici non singoalri ed assicurare una migliore stabilità al generico passo k-esimo si sceglie l'elemento pivot scegliendo una delle due strategie:

- Pivoting totale: si sceglie $a_{pq}^{(k-1)}$ con $p, q \in [k, n]$ tale che $a_{pq}^{(k-1)} = \max_{i,j \in [k,n]} a_{ij}^{(k-1)}$, e se $p \neq k$, si scambiano le righe k e p, e si scambiano le colonne k e q; il metodo di Gauss con pivot totale risulta stabile, ma in pratica, a causa degli $\frac{n^3}{3}$ confronti, il tempo di esecuzione è circa raddoppiato, e quindi non è competitivo;
- Pivoting parziale: si sceglie $a_{pk}^{(k-1)}$ con $p \in [k, n]$ tale che $a_{pk}^{(k-1)} = \max_{i \in [k,n]} a_{ik}^{(k-1)}$, e se $p \neq k$, si scambiano le righe k e p; la stabilità del metodo di Gauss con pivot parziale non è dimostrabile ma sperimentalmente accettata; per n grande il tempo di esecuzione è circa lo stesso di Gauss classico ($\frac{n^3}{3}$ confronti), e quindi risulta essere il metodo più usato (tra i metodi diretti).

É possibile vedere il metodo di Gauss con pivoting parziale in termini matriciali:

$$\begin{cases}[\mathrm{A}^{(0)}|b^{(0)}] = [\mathrm{A}|b] \\ [\mathrm{A}^{(k)}|b^{(k)}] = \mathrm{M}_k\mathrm{P}_k[\mathrm{A}^{(k-1)}|b^{(k-1)}], \forall k \in [1, n-1]\end{cases}$$

Dove $\forall k \in [1, n-1]\mathrm{P}_k := [e_1, e_2, ..., e_p, ..., e_k, ..., e_n]$. In altri termini:

$$\begin{cases}\mathrm{M}_{n-1}\mathrm{P}_{n-1}\mathrm{M}_{n-2}\mathrm{P}_{n-2}...\mathrm{M}_2\mathrm{P}_2\mathrm{M}_1\mathrm{P}_1\mathrm{A} = \mathrm{R} \\ \mathrm{P} := \mathrm{P}_1\mathrm{P}_2...\mathrm{P}_{n-2}\mathrm{P}_{n-1}\end{cases} \Rightarrow$$

$$\Rightarrow \begin{cases}\underbrace{(\mathrm{M}_{n-1}\mathrm{P}_{n-1}\mathrm{M}_{n-2}\mathrm{P}_{n-2}...\mathrm{M}_2\mathrm{P}_2\mathrm{M}_1\mathrm{P}_1)\mathrm{P}^{-1}}_{=\mathrm{L}^{-1}}\mathrm{PA} = \mathrm{R} \quad (\mathrm{PP}^{-1} = \mathrm{I}) \\ \mathrm{L} := \mathrm{P}(\mathrm{P}_1\mathrm{M}_1^{-1}\mathrm{P}_2\mathrm{M}_2^{-1}...\mathrm{P}_{n-1}\mathrm{M}_{n-1}^{-1}\mathrm{P}_{n-1}\mathrm{M}_{n-1}^{-1})\end{cases} \Rightarrow$$

$$\Rightarrow \mathrm{L}^{-1}\mathrm{PA} = \mathrm{R} \Rightarrow \mathrm{PA} = \mathrm{LR}$$

Il metodo di Gauss con pivot parziale si arresta se $\det(\mathrm{A}^{(k-1)}) \neq 0 \Leftrightarrow \det(\mathrm{A}) \neq 0$. Per il sistema lineare:

$$\mathrm{A}x = b \Leftrightarrow \mathrm{PA}x = \mathrm{P}b \Leftrightarrow \mathrm{LR}x = \mathrm{P}b \Leftrightarrow \begin{cases}\mathrm{L}y = \mathrm{P}b \\ \mathrm{R}x = y\end{cases}$$

$$\begin{aligned}&\mathrm{A}x = b \rightarrow \\ &\rightarrow \mathrm{M}_{n-1}\mathrm{P}_{n-1}\mathrm{M}_{n-2}\mathrm{P}_{n-2}...\mathrm{M}_2\mathrm{P}_2\mathrm{M}_1\mathrm{P}_1\mathrm{A}x = \\ &= \mathrm{M}_{n-1}\mathrm{P}_{n-1}\mathrm{M}_{n-2}\mathrm{P}_{n-2}...\mathrm{M}_2\mathrm{P}_2\mathrm{M}_1\mathrm{P}_1 b \rightarrow \\ &\rightarrow \mathrm{M}_{n-1}\mathrm{P}_{n-1}\mathrm{M}_{n-2}\mathrm{P}_{n-2}...\mathrm{M}_2\mathrm{P}_2\mathrm{M}_1\mathrm{P}_1\mathrm{A}x = \\ &= (\mathrm{M}_{n-1}\mathrm{P}_{n-1}\mathrm{M}_{n-2}\mathrm{P}_{n-2}...\mathrm{M}_2\mathrm{P}_2\mathrm{M}_1\mathrm{P}_1)\mathrm{P}^{-1}\mathrm{P}b \rightarrow \\ &\rightarrow \mathrm{R}x = \mathrm{L}^{-1}\mathrm{P}b\end{aligned}$$

E quindi il sistema lineare $\mathrm{L}y = \mathrm{P}b$ viene implicitamente risolto.

Metodo di eliminazione di Gauss con pivoting parziale

```
1: function ALGORITMOGAUSSPIVOTPARZIALE(A)
2:     for k = 1 to n do
3:         |a_pk| ← max_{i∈[k,n]} |a_ik|
4:         r_k ← p
5:         ∀j ∈ [k, n], scambia a_kj e a_pj
6:         ∀j ∈ [k, n], w_j ← a_kj
7:         for i = k + 1 to n do
8:             η = a_ik / a_kk
9:             a_ik ← η
10:            for j = k + 1 to n do
11:                a_ij ← a_ij − η w_j
12:            end for
13:        end for
14:    end for
15: end function
```

Costo computazionale

Richiede $\frac{n^3}{3}$ operazioni.

Osservazioni

Da notare che non si costruiscono esplicitamente le matrici di trasformazione $M_k P_k$. Inoltre è necessario controllare $|a_{pk}|$: poiché lavoriamo con numeri floating point, è preferibile verificare se $|a_{pk}| \leq toll$, dove $toll = \epsilon_m ||A||_\infty$.

7.10.1 Teorema

Sia A $\in M_{n*n}(\mathbb{R})$, se $det(A) \neq 0$, esiste una matrice di permutazione P tale che PA = LR alla Dolittle.

7.11 Metodi iterativi

Nei metodi diretti la presenza di eventuali elementi nulli nella matrice non può essere sfruttata ai fini di ridurre sia il costo computazionale, sia l'occupazione di memoria (che sono aspetti significativi per sistemi di grandi dimensioni). Infatti, la trasformazione di A può introdurre un numero diverso laddove prima c'era uno zero (fill in). I metodi iterativi sono utili per la risoluzione di sistemi lineari di grandi dimensioni con matrici A sparse (il numero degli elementi non nulli è dell'ordine di n). Dato il sistema lineare $Ax = b$, con $A \in M_{n*n}(\mathbb{R})$ e $b \in \mathbb{R}^n$, con soluzione $x^* \in \mathbb{R}^n$, si vuol costruire, partendo da un $x^{(0)} \in \mathbb{R}^n$, una successione $\{x^{(k)}\}$ tale che:

- $x^{(k)} \to x^*$;
- $x^{(k)}$ è facile e non dispendioso da costruire.

Due esempi di metodi iterativi sono:

- Il metodo di Jacobi (o metodo degli spostamenti simultanei):

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1, j\neq i}^{n} a_{ij} x_i^{(k)}}{a_{ii}}, a_{ii} \neq 0$$

- Il metodo di Gauss-Seidel (o metodo degli spostamenti successivi):

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_i^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_i^{(k)}}{a_{ii}}, a_{ii} \neq 0$$

  (Si può notare che nella prima sommatoria si usano le componenti "nuove" del vettore).

É possibile, per entrambi i metodi, dare una formulazione matriciale: sia A = L + D + U, sia ha che:

- Per il metodo di Jacobi $\mathrm{D}x^{(k+1)} = -(\mathrm{L}+\mathrm{U})x^{(k)} + b$;
- Per il metodo di Gauss-Seidel $(\mathrm{L}+\mathrm{D})x^{(k+1)} = -\mathrm{U}x^{(k)} + b$;

Jacobi e Gauss-Seidel sono casi particolari di "splitting" di A: si cercano due matrici M, N con M non singolare tali che $\mathrm{A} = \mathrm{M} - \mathrm{N}$, da cui

$$\mathrm{A}x = b \Leftrightarrow (\mathrm{M}-\mathrm{N})x = b \Leftrightarrow \mathrm{M}x = \mathrm{N}x + b \Leftrightarrow x = \mathrm{M}^{-1}\mathrm{N}x + \mathrm{M}^{-1}b$$

Da cui si ricava il procedimento iterativo $x^{(k+1)} = \mathrm{M}^{-1}\mathrm{N}x^{(k)} + \mathrm{M}^{-1}b$. Non è detto che per calcolare $x^{(k)}$ si calcoli effettivamente l'inversa di M, in generale si risolve il sistema lineare "facile" $\mathrm{M}x^{(k+1)} = \mathrm{N}x^{(k)} + b$

7.12 Metodo iterativo convergente

Un metodo iterativo per sistemi lineari $x^{(k+1)} = \mathrm{P}x^{(k)} + q$ (dove P è la matrice di iterazione) è convergente se $\forall x^{(0)}$, $\lim_{k\to\infty} ||e^{(k)}|| = 0$, dove $e^{(k)} = x^{(k)} - x^*$ e $x^* = \mathrm{P}x^* + q$ (x^* è la soluzione esatta).

7.12.1 Teorema

$\forall x^{(0)}, e^{(k)} = x^{(k)} - x^* = \mathrm{P}x^{(0)}$ converge a $\underline{0}$ se e solo se $\{P^{(k)}\}$ converge.

Corollario

$\{e^{(k)}\}$ converge a $\underline{0}$ se e solo se $\rho(\mathrm{P}) < 1$.

Corollario

$\{e^{(k)}\}$ converge a $\underline{0}$ se vale $||\mathrm{P}|| < 1$ per una qualunque norma naturale.

7.13 Velocità di convergenza

É importante sapere "quanto velocemente" converge il metodo. Si può tentare di stimare il numero k di iterazioni necessarie per ridurre l'errore iniziale di un fattore maggiore o uguale a 10^{-m}, ossia

$$||e^{(k)}|| \leq 10^{-m}||e^{(0)}|| \Leftrightarrow \frac{||e^{(k)}||}{||e^{(0)}||} \leq 10^{-m}$$
$$||e^{(k)}|| = ||\mathrm{P}^k e^{(0)}|| \leq ||\mathrm{P}^k||||e^{(0)}||$$

Per ogni norma naturale indotta dalla norma vettoriale. Si può scegliere k tale che $||\mathrm{P}^k||||e^{(0)}|| \leq 10^{-m}||e^{(0)}||$, ovvero $||\mathrm{P}^k|| \leq 10^{-m}$ tenendo conto che esiste una norma naturale tale che $\forall \epsilon > 0, \rho(\mathrm{P}^k) \leq ||\mathrm{P}^k|| \leq \rho(\mathrm{P}^k) + \epsilon$, si cerca k tale che $\rho(\mathrm{P}^k) = (\rho(\mathrm{P}))^k \leq 10^{-m}$, ovvero $k\log_{10}(\rho(\mathrm{P})) \leq -m$ da cui $k \leq \dfrac{m}{-\log_{10}(\rho(\mathrm{P}))}$. Si definisce $\mathrm{R} := -\log_{10}(\rho(\mathrm{P}))$ velocità di convergenza: maggiore è R (più piccolo è il raggio spettrale), minori sono le iterazioni necessarie.

7.13.1 Teorema

Il metodo di Jacobi converge se A è a diagonale dominante in senso forte oppure se A^T è a diagonale dominante in senso forte.

7.13.2 Teorema

Il metodo di Gauss-Seidel converge se A è a diagonale dominante in senso forte.

7.13.3 Teorema

Il metodo di Gauss-Seidel converge se A è simmetrica e definita positiva.

7.14 Teorema di Stein-Rosemberg

Sia A una matrice per cui $\forall i \in [1, n], a_{ii} \neq 0$, sia la relativa matrice $J = M^{-1}N$ del metodo di Jacobi con elementi maggiori o uguali a zero. Sia $GS = M^{-1}N$ la matrice relativa ad A del metodo di Gauss-Seidel, allora una sola di queste condizioni è valida:

- $\rho(GS) = \rho(J) = 1$;
- $\rho(GS) = \rho(J) = 0$;
- $\rho(GS) < \rho(J) < 1$;
- $\rho(GS) > \rho(J) > 1$.

Capitolo 8

Interpolazione

Nella pratica si presentano spesso i seguenti problemi:

- Da misure sperimentali so state ricavate le coppie di valori $\forall i \in [0,n], (x_i, f_i)$ che esprimono un campionamento di un fenomeno fisico, e supponendo tali valori esatti, si vuole conoscere il valore $\bar{y} = f(\bar{x})$ per $\forall i \in [0,n], \bar{x} \neq x_i$, e quindi ottenere una funzione che rappresenti il fenomeno;
- Si ha una funzione estremamente complicata, il cui calcolo richiede un elevato tempo macchina, e pertanto, si tabula la funzione in un prefissato numero di punti e si approssima mediante interpolazione.

In entrambi i casi, si suppone che i dati appartengano ad una funzione che non presenta difficoltà di calcolo (polinomiale o razionale) e li si utilizza per cercarla. In generale, date le coppie di valori $(x_0, f_0), (x_1, f_1), ..., (x_n, f_n)$ con $\forall i \in [0,n], x_i \in [a,b]$ (x_i punti fondamentali) e S uno spazio di funzioni definite su $[a,b]$; si cerca una funzione $g \in$ S che soddisfi le condizioni di interpolazione: $\forall i \in [0,n], g(x_i) = f_i$. Vogliamo valutare g in $\forall i \in [0,n], \bar{x} \neq x_i$: sia $\epsilon_1 = \min\{x_0, x_1, ..., x_2\}$ e $\epsilon_2 = \max\{x_0, x_1, ..., x_2\}$, parleremo di:

- Interpolazione se $\forall \bar{x} \in [\epsilon_1, \epsilon_2]$;
- Estrapolazione se $\forall \bar{x} \notin [\epsilon_1, \epsilon_2]$.

La scelta dello spazio di funzioni S cade su spazi lineari a dimensione finita $(n+1)$, pertanto si può fissare una base $\{\phi_j(x), \forall i \in [0,n]\}$ ed esprimere $g(x) \in$ S come $g(x) = \sum_{j=0}^{n} a_j \phi_j(x)$.

Trovare la funzione interpolante $g(x) \in$ S equivale a trovare $a_0, a_1, ..., a_n$ tali che

$$\forall i \in [0,n], g(x_i) = \sum_{j=0}^{n} a_j \phi_j(x_j) = f_i$$

$$\begin{cases} a_0\phi_0(x_0) + a_1\phi_1(x_0) + ... + a_n\phi_n(x_0) + = f_0 \\ a_0\phi_0(x_1) + a_1\phi_1(x_1) + ... + a_n\phi_n(x_1) + = f_1 \\ \vdots \\ a_0\phi_0(x_n) + a_1\phi_1(x_n) + ... + a_n\phi_n(x_n) + = f_n \end{cases}$$

$$\text{G} := \begin{bmatrix} \phi_0(x_0) & \phi_1(x_0) & \dots & \phi_n(x_0) \\ \phi_0(x_1) & \phi_1(x_1) & \dots & \phi_n(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(x_n) & \phi_1(x_n) & \dots & \phi_n(x_n) \end{bmatrix}, \text{ G matrice di Gram}$$

La scelta dello spazio di funzioni S dipende dalle applicazioni, ed è molto importante. Degli esempi di interpolazione sono:

- L'interpolazione polinomiale: $g(x) = \sum_{k=0}^{n} a_k x^k$;
- L'interpolazione trigonometrica: $g(x) = a_0 + a_1 \cos x + a_2 \sin x + ...$;
- L'interpolazione spline;

8.1 L'interpolazione polinomiale

$S = \mathbb{P}_n \Rightarrow \forall j \in [0,n], \phi_j(x) = x^j \Rightarrow g(x) = a_0 + a_1 x + ... + a_n x^n$. Si vuole determinare un polinomio di grado minore o uguale a n, ossia $p(x) \in \mathbb{P}_n$, tale che
$\forall i \in [0,n], p(x_i) = \sum_{j=0}^{n} a_j x^j = f_j$, ovvero

$$\begin{cases} a_0 + a_1 x_0 + a_2 x_0^2 ... + a_n x_0^n + = f_0 \\ a_0 + a_1 x_1 + a_2 x_1^2 ... + a_n x_1^n + = f_1 \\ a_0 + a_1 x_2 + a_2 x_2^2 ... + a_n x_2^n + = f_2 \\ \vdots \\ a_0 + a_1 x_n + a_2 x_n^2 ... + a_n x_n^n + = f_n \end{cases}$$

Il problema consiste nel risolvere un sistema lineare nelle incognite $a_0, a_1, ..., a_n$. La matrice di Gram assume la forma

$$\mathrm{G} := \begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix}, \text{ G matrice di Cauchy-Vandelmonde}$$

8.2 Polinomio interpolante di Lagrange

La matrice di Gram G ottenuta con la base dei monomi è fortemente malcondizionata, pertanto si può cambiare la base per $\mathbb{P}_n$. La situazione ottimale si ha per $G = I$, ovvero, occorrono $n+1$ funzioni $l_0^{(n)}(x), l_1^{(n)}(x), ..., l_n^{(n)}(x)$ tali che:

- $l_0^{(n)}(x), l_1^{(n)}(x), ..., l_n^{(n)}(x) \in \mathbb{P}_n$ sono linearmente dipendenti;
- $\forall i \in [0,n], \forall j \in [0,n], l_j^{(n)}(x_i) = \delta_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$

In generale funzioni di questo tipo assumono la forma

$$\forall i \in [0,n], l_i^{(n)}(x) = \frac{\prod_{j=0, j\neq i}^{n} (x - x_j)}{\prod_{j=0, j\neq i}^{n} (x_i - x_j)}$$

Dove $l_i^{(n)}(x)$ è detta funzione fondamentale di Lagrange. Si verifica facilmente che i polinomi $l_i^{(n)}(x)$ di grado n soddisfano $l_j^{(n)}(x_i) = \delta_{ij}$, e il polinomio di grado n interpolante $(x_0, f_0), (x_1, f_1), ..., (x_n, f_n)$ assume la forma

$$\mathcal{L}_n(x, f) = f_0 l_0^{(n)}(x) + f_1 l_1^{(n)}(x) + ... + f_n l_n^{(n)}(x)$$

Dove $\mathcal{L}_n(x, f)$ è detto polinomio interpolante di Lagrange. Essendo $\forall i \neq j, x_i \neq x_j$, tale polinomio esiste sempre ed è univocamente determinato. Le funzioni fondamentali $l_i^{(n)}(x)$ dipendono esclusivamente dei punti x_i (punti fondamentali), pertanto risultano univocamente determinate una volta fissati i punti x_i.

8.3 Polinomio interpolante di Newton

Il polinomio interpolante di Lagrange presenta difficoltà di applicazione se si vogliono alimentare le informazioni su $f(x)$, ovvero il numero di coppie $(x_0, f_0), (x_1, f_1), ..., (x_n, f_n)$. Poiché le funzioni fondamentali $l_i^{(n)}(x)$ dipendono da tutti i punti x_i, l'inserimento di un nuovo punto obbliga a ricostruire ex-novo tutte le funzioni fondamentali. Conviene esprimere in una forma diversa il polinomio interpolante, in modo da poter aggiungere dei punti senza modificare i calcoli precedenti. Sia $\mathcal{L}_n(x, f)$ il polinomio di grado al massimo n interpolante $(x_0, f_0), (x_1, f_1), ..., (x_n, f_n)$, e supponiamo di voler aggiungere la nuova coppia (x_{n+1}, f_{n+1}), si vuole ottenere $\mathcal{L}_{n+1}(x, f) \in \mathbb{P}_{n+1}$ tale che

- $\forall i \in [0, n+1], \mathcal{L}_{(n+1)}(x_i, f) = f_i$
- $\mathcal{L}_{n+1}(x, f) = \mathcal{L}_n(x, f) + \mathcal{Q}(x)$.

$\mathcal{Q}(x)$ deve:

- Essere un polinomio di grado $n + 1$;
- Assumere zero nei vecchi punti di interpolazione ($\forall i \in [0, n], \mathcal{Q}(x_i) = 0$).

Pertanto si esprime $\mathcal{Q}(x) = \mathcal{C}_{n+1} w_{n+1}(x) = \mathcal{C}_{n+1}(x - x_0)(x - x_1)...(x - x_n)$, da cui $\mathcal{L}_{n+1}(x, f) = \mathcal{L}_n(x, f) + \mathcal{C}_{n+1} w_{n+1}(x)$. Da $\mathcal{L}_{n+1}(x_{n+1}, f) = f_{n+1}$ si ottiene $\mathcal{C}_{n+1} = \dfrac{f_{n+1} - \mathcal{L}_{n+1}(x_{n+1}, f)}{w_{n+1}(x_{n+1})}$, ossia $\mathcal{C}_{n+1} w_{n+1}(x) = \mathcal{L}_{n+1}(x, f) - \mathcal{L}_n(x, f)$, pertanto

$$\begin{aligned}
\mathcal{L}_{n+1}(x, f) &= \mathcal{L}_0(x, f) - \mathcal{L}_0(x, f) + \mathcal{L}_1(x, f) - \mathcal{L}_1(x, f) + ... \\
&\quad ... + \mathcal{L}_{n-1}(x, f) - \mathcal{L}_{n-1}(x, f) + \mathcal{L}_n(x, f) - \mathcal{L}_n(x, f) = \\
&= \mathcal{L}_0(x, f) + (\mathcal{L}_1(x, f) - \mathcal{L}_0(x, f)) + (\mathcal{L}_2(x, f) - \mathcal{L}_1(x, f)) + ... \\
&\quad ... + (\mathcal{L}_n(x, f) - \mathcal{L}_{n-1}(x, f)) = f_0 + \sum_{k=1}^{n} \mathcal{C}_k w_k(x) = \mathcal{N}(x, f)
\end{aligned}$$

Dove $\mathcal{N}(x, f)$ è detto polinomio interpolante di Newton. Assegnata una funzione $f(x)$ definita (almeno) sui punti $x_0, x_1, ..., x_n$, si definiscono differenze divise del primo ordine relativamente agli x_i le quantità $f[x_i, x_{i+1}] = \dfrac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$, e ricorsivamente le differenze divise di ordine k:

$$f[x_i, x_{i+1}, ..., x_{i+k-1}, x_{i+k}] = \frac{f[x_{i+1}, ..., x_{i+k-1}, x_{i+k}] - f[x_i, x_{i+1}, ..., x_{i+k-1}]}{x_{i+k} - x_i}$$

E per definizione la differenza di ordine 0 $f[x_i] = f(x_i)$. Si può dimostrare che $f[x_0, x_1, ..., x_k] = \mathcal{C}_k$, e quindi $\mathcal{N}(x, f) = f_0 + \sum_{k=1}^{n} f[x_0, x_1, ..., x_k] w_k(x)$.

Capitolo 9

Errore di interpolazione

Per costruzione e in assenza di errori di arrotondamento, dati
$\forall i \in [0,n], (x_i, f_i) : x_i \in [a,b]$ il polinomio interpolante soddisfa $\mathcal{L}_n(x_i, f) = f_i$, ossia:

- Se si conoscono solo i valori (x_i, f_i) non ha senso porsi il problema dello studio dell'errore;
- Se $f_i = f(x_i)$ con $f(x)$ funzione nota in forma esplicita, studiare l'errore significa esaminare il comportamento del polinomio $\mathcal{L}_n(x, f)$ rispetto alla funzione $f(x)$ in x punti distinti dai punti x_i, ovvero studiare la funzione errore $e(x) = f(x) - \mathcal{L}_n(x, f)$.

Poiché $\mathcal{L}_n(x, f)$ interpola $f(x)$ nei punti x_i, si ha

$$\forall i \in [0,n], e(x_i) = 0 \Rightarrow e(x) = (x - x_0)(x - x_1)...(x - x_n)\mathcal{R}(x) \Rightarrow$$
$$\Rightarrow e(x) = w_{n+1}(x)\mathcal{R}(x)$$

Se $f \in \mathrm{C}^{n+1}[a,b]$ si può dare una valutazione di $\mathcal{R}(x)$. Sia $f \in \mathrm{C}^{n+1}[a,b]$ e sia
$\mathrm{F}(t) = f(t) - \mathcal{L}_n(t, f) - w_{n+1}(t)\mathcal{R}(t)$ e $\epsilon_1 = \min\{x_0, x_1, ..., x_2\}$, $\epsilon_1 = \max\{x_0, x_1, ..., x_2\}$.
$\mathrm{F}(t)$ si annulla per $\forall i \in [0,n], t = x_i$ e $t = x$ (ossia ha $n+2$ zeri). Per il teorema di Rolle:

- Esistono almeno $n+1$ punti in (ϵ_1, ϵ_2) in cui si annulla $\mathrm{F}'(t)$;
- Esistono almeno n punti in (ϵ_1, ϵ_2) in cui si annulla $\mathrm{F}''(t)$;
- ...
- Esiste almeno 1 punto in (ϵ_1, ϵ_2) in cui si annulla $\mathrm{F}^{n+1}(t)$;

$\mathrm{F}^{n+1}(\epsilon) = f^{(n+1)}(\epsilon) - (n+1)!\,\mathcal{R}(x) \Rightarrow \mathcal{R}(x) = \dfrac{f^{(n+1)}(\epsilon)}{(n+1)!}$. Se $\forall x \in [a,b]$ si ha
$|f^{(n+1)}| \leq M \Rightarrow |\mathcal{R}(x)| \leq \dfrac{M}{(n+1)!}$. Se $f \notin \mathrm{C}^{n+1}[a,b]$ si può dare una valutazione di $\mathcal{R}(x)$ in altri termini. Sia $\forall i \in [0,n], (x_{n+1}, f_{n+1}) = (x, f(x))$ con $x \neq x_i$ si ha
$\mathcal{L}_{n+1}(t, f) = \mathcal{L}_n(t, f) + f[x_0, x_1, ..., x_n, t]w_{n+1}(t)$. Per $t = x$:

$$\mathcal{L}_{n+1}(x, f) = \mathcal{L}_n(x, f) + f[x_0, x_1, ..., x_n, x]w_{n+1}(x) \Rightarrow$$
$$\Rightarrow e(x) = f(x) - \mathcal{L}_n(x, f) = f[x_0, x_1, ..., x_n, x]w_{n+1}(x)$$

Se $f \in \mathrm{C}^{n+1}[a,b] \Rightarrow f[x_0, x_1, ..., x_n, x] = \dfrac{f^{n+1}(\epsilon)}{(n+1)!}, \epsilon \in (\epsilon_1, \epsilon_2)$. In conclusione:

$$e(x) = w_{n+1}(x)\mathcal{R}(x), \mathcal{R}(x) = \begin{cases} \frac{f^{n+1}(\epsilon)}{(n+1)!} \text{ se } f \in \mathrm{C}^{n+1}[a,b] \\ f[x_0, x_1, ..., x_n, x] \text{ se } f \notin \mathrm{C}^{n+1}[a,b] \end{cases}$$

Da cui abbiamo che:

- $\mathcal{R}(x)$ dipende dai punti e dalla $f(x)$;
- $w_{n+1}(x)$ dipende solo dai punti x_i, e quindi si possono determinare i punti x_i in modod che risulti minimo $|w_{n+1}(x)|$ qualunque sia x e ridurre così l'errore di interpolazione.