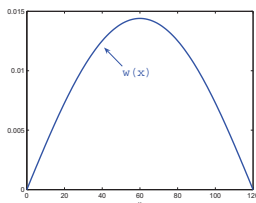


Supponendo che il carico sia lineare $t(x) = x$ e che la trave sia di acciaio e abbia le seguenti caratteristiche:

$$L = 120 \text{ cm} \quad q = 100 \text{ kg/m} \quad E = 3.0 \times 10^7 \text{ kg/cm}^2$$

$$S = 1000 \text{ kg} \quad I = 625 \text{ cm}^4$$

approssimare la deflessione $w(x)$ ogni 6 cm con uno **schema alle differenze finite**.



Soluzione esatta

Esempio 1

Una trave di lunghezza L , **fissata** agli estremi e tesa lungo la direzione della sua lunghezza da una **forza** S , è soggetta a un **carico trasversale** $t(x)dx$. Nell'ipotesi di piccole deflessioni, la **deflessione** della trave $w(x)$ è descritta dal **problema differenziale** lineare ai limiti del secondo ordine

$$\begin{cases} \frac{d^2w}{dx^2} = \frac{S}{EI}w(x) + \frac{q}{2EI}t(x)(x-L) & 0 < x < L \\ w(0) = w(L) = 0. \end{cases}$$

x : **distanza** dall'estremo sinistro della trave; L : **lunghezza** della trave; $t(x)$: intensità del **carico**; E : **modulo di elasticità**; S : **tensione** agli estremi della trave; I : **momento principale di inerzia**

Soluzione

La soluzione approssimata $\{y_i \approx y(x_i)\}_{i=1, \dots, N+1}$, nei nodi $x_i = a + ih$, $i = 1, \dots, N$, si ottiene risolvendo il **sistema lineare tridiagonale** di N equazioni nelle N incognite $Y = [y_1, y_2, \dots, y_N]^T$

$$AY = B$$

con

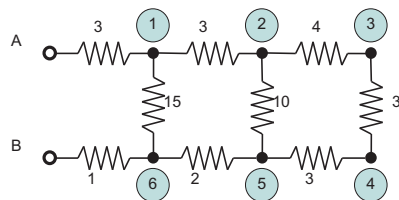
$$A = \begin{bmatrix} 2 + h^2 \frac{S}{EI} & -1 & 0 & \dots & \dots & 0 \\ -1 & 2 + h^2 \frac{S}{EI} & -1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 0 & -1 & 2 + h^2 \frac{S}{EI} \end{bmatrix}$$

$$B = [h^2 r(x_1) + \alpha, h^2 r(x_2), \dots, h^2 r(x_{N-1}), h^2 r(x_N) + \beta]^T$$

$$r(x) = \frac{q}{2EI} x(x-L)$$

Esempio 2

Determinare i potenziali nei nodi 1 – 6 del circuito (le resistenze sono misurate in *Ohm*)



Soluzione

Applicando la **legge di Kirchoff** $\sum_i I_i = 0$ in ogni nodo si ottiene il sistema lineare

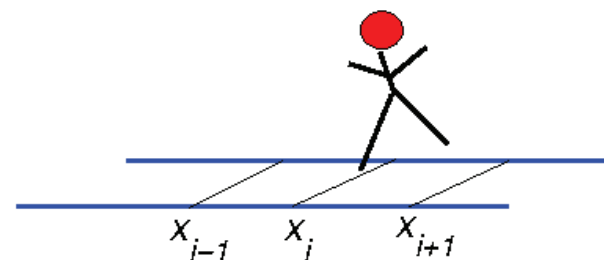
$$\begin{cases} 11v_1 & -5v_2 & & & & -v_6 & = & 500 \\ -20v_1 & +41v_2 & -15v_3 & & & -6v_5 & = & 0 \\ & -3v_2 & +7v_3 & -4v_4 & & & = & 0 \\ & & -v_3 & +2v_4 & -v_5 & & = & 0 \\ & & & -10v_4 & +28v_5 & -15v_6 & = & 0 \\ -2v_1 & & & -15v_5 & +47v_6 & = & 0 \end{cases}$$

4

Esempio 3

Un ubriaco compie una **passeggiata casuale**, facendo un passo a sinistra o a destra a caso lungo una strada rettilinea. Quando raggiunge una estremità della strada, si ferma.

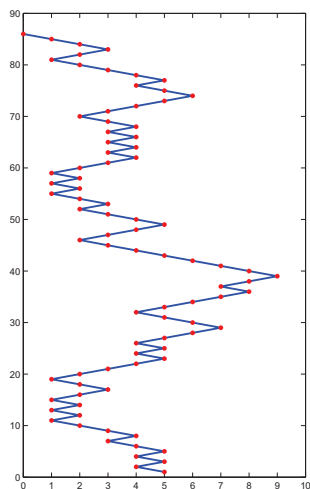
Calcolare la **probabilità** che l'ubriaco raggiunga l'estremità sinistra della strada partendo dalla posizione i .



5

Esempio di passeggiata per $N = 10$

Si può simulare una **passeggiata casuale** tirando una moneta.



Punto di partenza: x_5

6

Soluzione

La **probabilità** p_i , $i = 0, 1, \dots, N$, di raggiungere l'estremo sinistro partendo dalla posizione i , soddisfa la relazione

$$\begin{cases} p_0 = 1 & p_N = 0 \\ p_i = \frac{1}{2}p_{i-1} + \frac{1}{2}p_{i+1} & i = 1, \dots, N-1 \end{cases}$$

Si tratta di un **sistema lineare tridiagonale** nelle incognite

$$p_i, i = 1, \dots, N-1$$

$$\begin{cases} p_1 & -\frac{1}{2}p_2 & & & & = & \frac{1}{2} \\ -\frac{1}{2}p_1 & +p_2 & -\frac{1}{2}p_3 & & & = & 0 \\ \dots & \dots & \dots & \dots & \dots & = & 0 \\ & & & -\frac{1}{2}p_{N-3} & +p_{N-2} & -\frac{1}{2}p_{N-1} & = & 0 \\ & & & & -\frac{1}{2}p_{N-2} & +p_{N-1} & = & 0 \end{cases}$$

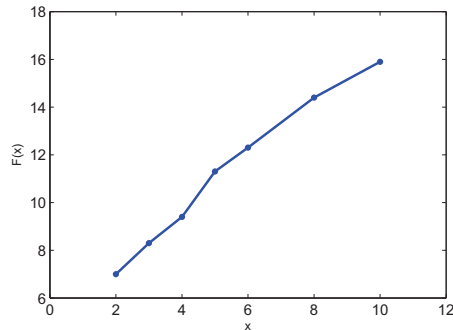
7

Esempio 4

Nella tabella sono riportate le **misure sperimentali** relative alla **forza** $F(x)$ necessaria per allungare una **molla** fino alla **lunghezza** x .

x	2	3	4	5	6	8	10
$F(x)$	7.0	8.3	9.4	11.3	12.3	14.4	15.9

Determinare la **costante elastica** della molla.



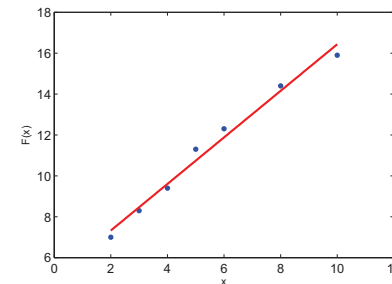
8

Soluzione

La forza $F(x)$ necessaria per allungare una molla fino alla lunghezza x è data da $F(x) = k(x - l)$ (**Legge di Hooke**) dove k è la **costante elastica** e l è la **lunghezza a riposo** della molla.

Per trovare la costante elastica della molla si devono approssimare i dati in tabella con il **polinomio ai minimi quadrati** di primo grado (**retta di regressione**): $P_1(x) = a_0 + a_1 x$

Il coefficiente a_1 fornisce l'approssimazione della **costante elastica**.



```
>> X=[2, 3, 4, 5, 6, 8, 10]
>> F=[7.0, 8.3, 9.4, 11.3, 12.3, 14.4, 15.9]
>> X2 = linspace(X(1),X(7))
>> p = polyfit(X,F,1);y2 = polyval(p,X2)
>> p

p =

1.1383 5.0491
```

9

Sistema delle equazioni normali

I coefficienti a_1, a_2 del polinomio ai minimi quadrati si ottengono risolvendo il **sistema delle equazioni normali**

$$\begin{cases} a_0 s_0 + a_1 s_1 = v_0 \\ a_0 s_1 + a_1 s_2 = v_1 \end{cases} \Rightarrow \begin{cases} a_0 = \frac{v_0 s_2 - v_1 s_1}{s_0 s_2 - s_1^2} \\ a_1 = \frac{s_0 v_1 - s_1 v_0}{s_0 s_2 - s_1^2} \end{cases}$$

dove

$$s_0 = n + 1 \quad s_1 = \sum_{i=0}^n x_i \quad s_2 = \sum_{i=0}^n x_i^2$$

$$v_0 = \sum_{i=0}^n F(x_i) \quad v_1 = \sum_{i=0}^n x_i F(x_i)$$

$$\Rightarrow a_0 = 5.049 \quad a_1 = 1.1383.$$

10

Metodi diretti per la soluzione di sistemi lineari

Sono basati sulla **trasformazione** del sistema di partenza in uno **equivalente** che abbia una struttura particolarmente **semplice** per cui è facile calcolarne la soluzione.

La **soluzione numerica** viene calcolata in un **numero finito** di passi e, se non vi fossero errori di arrotondamento nei dati o durante i calcoli, la soluzione numerica sarebbe **esatta**.

Data l'**occupazione di memoria** (RAM) richiesta nei passaggi dell'algoritmo, vengono utilizzati quando la **matrice dei coefficienti ha dimensione non "troppo" elevata**.

11

Costo computazionale di un algoritmo

Prima di implementare un algoritmo bisogna stimare il suo **costo computazionale**, cioè il numero di **operazioni pesanti** (moltiplicazioni o divisioni) necessarie per calcolare **numericamente** la soluzione.

Costo computazionale:

$$C_c \approx \text{numero di moltiplicazioni o divisioni}$$

Metodo di Cramer

Dall'**Algebra** sappiamo che la soluzione **esatta** di un sistema lineare si può ottenere con il **metodo di Cramer**.

Se A è **regolare**, $X = A^{-1}B \implies x_i = \frac{D_i}{\det A}$, $i = 1, 2, \dots, n$, dove D_i è il determinante della matrice ottenuta da A sostituendo alla colonna i -esima il vettore B .

12

Costo computazionale del metodo di Cramer

$$x_i = \frac{D_i}{\det A} \quad i = 1, 2, \dots, n$$

- $n + 1$ **determinanti** (D_i , $i = 1, \dots, n$ e $\det A$)

↓

- $n!$ **prodotti** per ciascun determinante

↓

- $n - 1$ **moltiplicazioni** per ciascun prodotto

+ n **divisioni** (trascurabili)

Costo computazionale:

$$C_c = (n + 1)n!(n - 1) + n \simeq (n + 1)n!(n - 1)$$

$$n = 15 \rightarrow C_c \simeq 3 \cdot 10^{14} \text{ moltiplicazioni} \rightarrow \text{circa 4 giorni}$$

$$n = 20 \rightarrow C_c \simeq 3 \cdot 10^{21} \text{ moltiplicazioni} \rightarrow \text{circa 4800 anni}$$

} **Inutilizzabile!!**

(supponendo $0.5 \cdot 10^{-10}$ secondi per operazione)

13

Metodo di eliminazione di Gauss

Il **metodo di eliminazione di Gauss** trasforma, in $n - 1$ passi, il sistema lineare

$$AX = B \quad X, B \in \mathbb{R}^n \quad A \in \mathbb{R}^{n \times n}$$

con matrice dei coefficienti A "**piena**", nel sistema **equivalente**

$$UX = \tilde{B} \quad X, \tilde{B} \in \mathbb{R}^n \quad U \in \mathbb{R}^{n \times n}$$

con matrice dei coefficienti U **triangolare superiore**.

Il metodo utilizza le seguenti **operazioni "lecite"**:

- **scambio** di 2 equazioni
- **somma** di un'equazione con un'altra **moltiplicata** per una costante

14

Fattorizzazione LU

Il **metodo di eliminazione di Gauss** può essere interpretato come la **fattorizzazione** della matrice di partenza A nel prodotto di due matrici triangolari.

Teorema. Se la matrice $A \in \mathbb{R}^{n \times n}$ ha **determinanti principali di testa** tali che

$$\det A_k \neq 0, \quad k = 1, 2, \dots, n,$$

allora

$$A = LU$$

dove $L \in \mathbb{R}^{n \times n}$ è una **matrice triangolare inferiore** con elementi diagonali pari a 1 e $U \in \mathbb{R}^{n \times n}$ è una **matrice triangolare superiore**.

15

Applicazioni della fattorizzazione

Soluzione di un sistema lineare

Consideriamo il **sistema lineare** $AX = B$ e supponiamo che la matrice dei coefficienti A possa essere **fattorizzata**.

$$AX = B \xrightarrow{A=LU} LUX = B \Rightarrow \begin{cases} LY = B & \text{sistema triang. inf.} \\ & \text{(algoritmo di sost. in avanti)} \\ UX = Y & \text{sistema triang. sup.} \\ & \text{(algoritmo di sost. all'indietro)} \end{cases}$$

Una volta **fattorizzata** A , la soluzione del sistema si ottiene risolvendo i due sistemi triangolari con **costo computazionale** $2\frac{n^2}{2}$.

20

Calcolo del determinante di A

$$\det A = \det(LU) = \det L \det U = \underbrace{\left(\prod_{k=1}^n l_{kk} \right)}_{\substack{\downarrow \\ \text{Teorema di Binet}}} \underbrace{\left(\prod_{k=1}^n u_{kk} \right)}_{\substack{\downarrow \\ = 1}} = a_{11}^{(1)} a_{22}^{(2)} \cdots a_{nn}^{(n)}$$

Se durante la fattorizzazione sono stati fatti s **scambi** di righe, allora

$$\det A = (-1)^s a_{11}^{(1)} a_{22}^{(2)} \cdots a_{nn}^{(n)}$$

22

Soluzione di più sistemi lineari

Se dobbiamo risolvere più sistemi lineari

$$AX_i = B_i \quad i = 1, 2, \dots, r$$

aventi la **stessa matrice** A e **diversi termini noti**, si fattorizza una volta per tutte la matrice $A = LU$ e si risolvono, per ogni vettore B_i , i due sistemi triangolari

$$\begin{cases} LY_i = B_i & i = 1, 2, \dots, r \\ UX_i = Y_i \end{cases}$$

per la soluzione dei quali il **costo computazionale** è "solo" n^2 .

21

Calcolo dell'inversa di A

La matrice inversa A^{-1} di una matrice **regolare** A è la matrice tale che

$$A A^{-1} = I \quad I : \text{matrice identità}$$

$$I := \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} = [E_1 \ E_2 \ \cdots \ E_n] \quad E_i = [0, 0, \dots, \underbrace{1}_i, 0, \dots, 0]^T$$

Vettori della **base canonica**

ica

$$A A^{-1} = I \Rightarrow A X_i = E_i \quad i = 1, 2, \dots, n \Rightarrow A^{-1} = [X_1 \ X_2 \ \cdots \ X_n]$$

Una volta nota la **fattorizzazione** di A , basta risolvere gli n sistemi

$$L Y_i = E_i, \quad U X_i = Y_i, \quad i = 1, \dots, n$$

$$\text{Costo computazionale: } C_c \simeq \frac{n^3}{3} + n \cdot n^2 = \frac{4}{3}n^3$$

23

Calcolo del rango di A

- Se l'algoritmo di eliminazione applicato alla **matrice quadrata** $A \in \mathbb{R}^{n \times n}$ termina regolarmente dopo $n - 1$ passi \Rightarrow la matrice A ha **rango massimo** pari a n (**matrice regolare**).
- Se l'algoritmo di eliminazione applicato alla **matrice quadrata** $A \in \mathbb{R}^{n \times n}$ non può proseguire dopo q passi perché $a_{rk}^{(k)} = 0$, $r = k, \dots, n \Rightarrow$ la matrice A ha **rango** pari a $q \leq n$ (**matrice singolare**).
- L'algoritmo di eliminazione applicato alla **matrice rettangolare** $A \in \mathbb{R}^{m \times n}$ termina necessariamente dopo $q \leq m$ passi \Rightarrow la matrice A ha **rango** pari a q .

Nota: le operazioni "lecite" conservano il rango.

24

Matrici tridiagonali

Nel caso in cui la matrice dei coefficienti A è **tridiagonale** la **fattorizzazione LU** è molto **semplice** in quanto le matrici L e U hanno una struttura semplice.

$$A = \begin{bmatrix} d_1 & s_1 & 0 & \cdots & 0 \\ a_2 & d_2 & s_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & d_{n-1} & s_{n-1} \\ 0 & 0 & \cdots & a_n & d_n \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \alpha_2 & 1 & 0 & \cdots & 0 \\ 0 & \alpha_3 & 1 & \cdots & 0 \\ \vdots & \vdots & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_n & 1 \end{bmatrix} \begin{bmatrix} u_1 & v_1 & 0 & \cdots & 0 \\ 0 & u_2 & v_2 & \cdots & 0 \\ \vdots & \vdots & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{n-1} & v_{n-1} \\ 0 & 0 & \cdots & 0 & u_n \end{bmatrix} = LU$$

25

Algoritmo di Thomas

$$\begin{cases} u_1 = d_1 \\ v_i = s_i & i = 1, 2, \dots, n-1 \\ \alpha_i = a_i/u_{i-1} & i = 2, 3, \dots, n \\ u_i = d_i - \alpha_i v_{i-1} & i = 2, 3, \dots, n \end{cases}$$

Costo computazionale: $C_c = \underbrace{(n-1)}_{\text{divisioni}} + \underbrace{(n-1)}_{\text{moltiplicazioni}} = 2n - 2$

Soluzione del sistema lineare $AX = B$

$$y_1 = b_1 \quad y_i = b_i - \alpha_i y_{i-1} \quad i = 2, 3, \dots, n$$

$$x_n = y_n/u_n \quad x_i = (y_i - v_i x_{i+1})/u_i \quad i = n-1, \dots, 1$$

Costo computazionale: $C_c = \underbrace{(n-1)}_{\text{moltiplicazioni}} + \underbrace{(n-1)}_{\text{moltiplicazioni}} + \underbrace{n}_{\text{divisioni}} = 3n - 2$

2

26

Esempio 1: soluzione

La **soluzione esatta** è $\bar{X} = [\bar{p}_0, \dots, \bar{p}_N]^T$, dove $\bar{p}_i = 1 - \frac{i}{N}$, $i = 0, \dots, N$.

Con il **solutore di Matlab** si ha:

N	$\ \bar{X} - X\ _\infty$	decimali esatti	$C_c \approx$	tempo di calcolo	occupazione di memoria
11	3.33e-016	15	333	0.000304 s	0.8 Kbyte
21	1.55e-015	14	2667	0.000356 s	3.2 Kbyte
51	4.16e-015	14	41667	0.000402 s	20 Kbyte
101	2.14e-014	13	333333	0.001555 s	8 Kbyte
501	1.11e-013	12	41666666	0.070898 s	200 Kbyte
5001	1.84e-012	11	41666666666	29.306639 s	2 Mbyte
10001	1.38e-011	10	333333333333	225.971643 s	800 Mbyte

Nota. In realtà gli elementi diversi da zero della matrice A sono

$$N + (N-1) + (N-1) = 3N - 2 \ll N^2$$

$\rightarrow A$ è una matrice **sparsa**

27

Istruzioni MATLAB

```
N = 10
A = diag(ones(N-1,1));
for i=2:N-1, A(i,i-1) = -0.5; end
for i=1:N-2, A(i,i+1) = -0.5; end
B = zeros(N-1,1);B(1)=0.5
tic
X = A\B
toc
Xe = 1-(1:N-1)/N
max(abs(X-Xe'))
```

Nota. tic e toc sono le istruzioni per calcolare il **tempo di esecuzione**.

Esercizio

Scrivere un programma (Matlab/Fortran/C) che risolva un sistema lineare con l'**algoritmo di Thomas** e utilizzarlo per risolvere il problema della passeggiata casuale.

Di quanto si riducono l'occupazione di memoria e il tempo di esecuzione rispetto al solutore di Matlab? Si riduce anche l'errore di propagazione?

28

Esercizio

Data la matrice tridiagonale $A = \begin{bmatrix} 2 & -2 & 0 \\ -2 & 3 & -1 \\ 0 & -1 & 4 \end{bmatrix}$

calcolarne l'inversa con l'algoritmo di Thomas.

Soluzione

Poiché $\det A = 6 \neq 0$, la matrice A è **regolare** e quindi ammette l'inversa.

L'inversa è definita dalla relazione $AA^{-1} = I$, pertanto le colonne di $A^{-1} = [X_1 \ X_2 \ X_3]$ sono le soluzioni dei sistemi lineari $AX_i = E_i$, dove E_i , $i = 1, 2, 3$, sono i tre vettore della base canonica in \mathbf{R}^3 .

29

Dall'uguaglianza $A = LU$, dove

$$L = \begin{bmatrix} 1 & 0 & 0 \\ \alpha_2 & 1 & 0 \\ 0 & \alpha_3 & 1 \end{bmatrix} \quad U = \begin{bmatrix} u_1 & v_1 & 0 \\ 0 & u_2 & v_2 \\ 0 & 0 & u_3 \end{bmatrix}$$

si ottiene

$$u_1 = 2 \quad v_1 = -2 \quad v_2 = -1$$

$$\alpha_2 = -1 \quad u_2 = 3 - (-1)(-2) = 1$$

$$\alpha_3 = -1 \quad u_3 = 4 - (-1)(-1) = 3$$

$$\Rightarrow L = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \quad U = \begin{bmatrix} 2 & -2 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 3 \end{bmatrix}$$

30

La soluzione dei sistemi lineari

$$LY_i = E_i \quad UX_i = Y_i$$

con gli algoritmi di sostituzione in avanti e indietro dà

$$X_1 = \begin{bmatrix} \frac{11}{6} \\ \frac{4}{3} \\ \frac{1}{3} \end{bmatrix} \quad X_2 = \begin{bmatrix} \frac{4}{3} \\ \frac{4}{3} \\ \frac{1}{3} \end{bmatrix} \quad X_3 = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix}$$

$$\Rightarrow A^{-1} = \begin{bmatrix} \frac{11}{6} & \frac{4}{3} & \frac{1}{3} \\ \frac{4}{3} & \frac{4}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$$

31

Sistemi non lineari

$$F(X) = 0$$

$$X, F \in \mathbb{R}^n$$

$$\rightarrow F(X) = AX - B$$

Sistemi lineari

$$AX = B$$

$$X, B \in \mathbb{R}^n \quad A \in \mathbb{R}^{n \times n}$$



$$X = [x_1, x_2, \dots, x_n]^T$$

$$F = [f_1(X), f_2(X), \dots, f_n(X)]^T$$

$$0 = [0, 0, \dots, 0]^T$$

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \dots \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{cases}$$

$$X = [x_1, x_2, \dots, x_n]^T$$

$$A = [a_{ij}]_{i,j=0}^n$$

$$B = [b_1, b_2, \dots, b_n]^T$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases}$$

Metodo del punto unito

Sistemi non lineari

$$F(X) = 0 \Leftrightarrow X = \Phi(X) \text{ con } \Phi = [\varphi_1(X), \varphi_2(X), \dots, \varphi_n(X)]^T$$

Se $\bar{X} \in \mathbb{R}^n$ è **radice** di F allora è **punto unito** di Φ :

$$F(\bar{X}) = 0 \Leftrightarrow \bar{X} = \Phi(\bar{X})$$

Sistemi lineari

$$AX = B \Leftrightarrow X = CX + Q \text{ con } Q = [q_1, q_2, \dots, q_n]^T$$

$$C = [c_{ij}]_{i,j=0}^n$$

Se $\bar{X} \in \mathbb{R}^n$ è **soluzione** di $AX = B$ allora è **punto unito** di $\Phi = CX + Q$:

$$A\bar{X} = B \Leftrightarrow \bar{X} = C\bar{X} + Q$$

Metodi iterativi a un punto

Il **punto unito** $\bar{X} = \Phi(\bar{X})$, $\bar{X} = [\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n]^T$, può essere **approssimato** generando la successione

$$\begin{cases} X^{(0)} \text{ dato} \\ X^{(k)} = \Phi(X^{(k-1)}) \\ k = 1, 2, \dots \end{cases} \Leftrightarrow \begin{cases} X^{(0)} = [x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}]^T \text{ dato} \\ x_1^{(k)} = \varphi_1(x_1^{(k-1)}, x_2^{(k-1)}, \dots, x_n^{(k-1)}) \\ x_2^{(k)} = \varphi_2(x_1^{(k-1)}, x_2^{(k-1)}, \dots, x_n^{(k-1)}) \\ \dots \\ x_n^{(k)} = \varphi_n(x_1^{(k-1)}, x_2^{(k-1)}, \dots, x_n^{(k-1)}) \end{cases}$$

Le funzione φ_i sono chiamate **funzioni di iterazione**.

Metodi iterativi per sistemi lineari

Nel caso **lineare** il **punto unito** $\bar{X} = [\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n]^T$, può essere **approssimato** generando la successione

$$\begin{cases} X^{(k)} = C X^{(k-1)} + Q \quad k = 1, 2, \dots \\ X^{(0)} \in \mathbb{R}^n \text{ dato} \end{cases}$$

$$\Rightarrow x_i^{(k)} = \sum_{j=1}^n c_{ij} x_j^{(k-1)} + q_i \quad i = 1, \dots, n$$

La matrice $C \in \mathbb{R}^{n \times n}$ è chiamata **matrice di iterazione**.

Convergenza

Per poter definire la **convergenza** di un metodo iterativo dobbiamo prima di tutto definire l'**errore di troncamento**

Errore di troncamento: $E^{(k)} = \bar{X} - X^{(k)} \in \mathbb{R}^n$

\swarrow \searrow
soluzione **soluzione**
esatta **approssimata**

Per "misurare" la lunghezza di un vettore $V \in \mathbb{R}^n$ si ricorre alla **norma di vettore**:

$$\|V\| = \left(\sum_{i=1}^n |v_i|^p \right)^{1/p}$$

Convergenza: $\lim_{k \rightarrow \infty} \|E^{(k)}\| = 0 \iff \lim_{k \rightarrow \infty} X^{(k)} = \bar{X}$

Se il metodo iterativo è **convergente**, in assenza di errori di arrotondamento si ottiene la **soluzione esatta** dopo un **numero infinito** di passi.

Nota. In pratica ci si arresta quando $\|E^{(k)}\| \leq \epsilon$ (**criterio di arresto**)

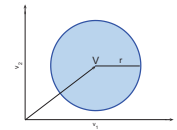
36

Norma di vettore

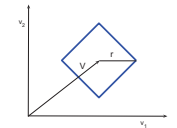
La **norma** di un vettore $V = [v_1, \dots, v_n]^T$ viene utilizzata per "misurare" la sua **lunghezza**.

Intorno: $\|V - W\| \leq r$

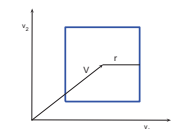
• **Norma due o euclidea:** $\|V\|_2 := \sqrt{\sum_{i=1}^n |v_i|^2}$



• **Norma uno:** $\|V\|_1 := \sum_{i=1}^n |v_i|$



• **Norma infinito:** $\|V\|_\infty := \max_{1 \leq i \leq n} |v_i|$



Nota. Tutte le norme sono **equivalenti**: $m\|V\|_p \leq \|V\|_q \leq M\|V\|_p$

37

Proprietà della norma di vettore

- $\|V\| \geq 0, \quad \|V\| = 0 \iff V = 0$
- $\|\alpha V\| = |\alpha| \cdot \|V\| \quad \forall \alpha \in \mathbb{R}, \forall V \in \mathbb{R}^n$
- $\|V + W\| \leq \|V\| + \|W\| \quad \forall V, W \in \mathbb{R}^n$ (*disuguaglianza triangolare*)

Distanza: in uno **spazio vettoriale normato** S è possibile introdurre la **distanza** tra due punti V e W in S

$$d(V, W) := \|V - W\|$$

Proprietà della distanza:

- $d(V, W) = 0 \iff V = W$
- $d(V, W) = d(W, V) \quad \forall V, W \in S$
- $d(V, W) \leq d(V, Z) + d(Z, W) \quad \forall V, W, Z \in S$

38

Norme di matrici

La **norma** di una matrice $A \in \mathbb{R}^{n \times n}$ soddisfa le seguenti

Proprietà

- $\|A\| \geq 0, \quad \|A\| = 0 \iff A = 0$
- $\|\alpha A\| = |\alpha| \cdot \|A\|, \quad \forall \alpha \in \mathbb{R}, \forall A \in \mathbb{R}^{n \times n}$
- $\|A + B\| \leq \|A\| + \|B\|, \quad \forall A, B \in \mathbb{R}^{n \times n}$ (*disuguaglianza triangolare*)
- $\|A \cdot B\| \leq \|A\| \cdot \|B\|, \quad \forall A, B \in \mathbb{R}^{n \times n}$

Definizione. Una matrice si dice **convergente** se $\lim_{k \rightarrow \infty} \|A^k\| = 0$

39

Norme indotte dalla norma di vettore

Ogni **norma di vettore** può essere utilizzata per definire una **norma di matrice** che permette di "misurare" come la matrice agisce sui vettori:

$$\|A\| = \max_{\|X\|=1} \|AX\| \quad A \in \mathbb{R}^{n \times n} \quad X \in \mathbb{R}^n$$

Le norme indotte soddisfano tutte le **proprietà delle norme** e, inoltre, soddisfano la **relazione di compatibilità** :

$$\|AX\| \leq \|A\| \cdot \|X\|$$

Infatti, se $X \neq 0$, si ha

$$\|A\| = \max_{\|X\|=1} \|AX\| = \max_{\|X\| \neq 0} \left\| \frac{AX}{\|X\|} \right\| = \max_{\|X\| \neq 0} \frac{\|AX\|}{\|X\|} \implies \|A\| \geq \frac{\|AX\|}{\|X\|}$$

Nota. Per tutte le norme indotte si ha $\|I\| = 1$ (I : matrice identità)

40

Norme indotte: esempi

• **Norma uno:** $\|A\|_1 := \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|$ (per colonne)

• **Norma infinito:** $\|A\|_\infty := \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|$ (per righe)

• **Norma due o spettrale:** $\|A\|_2 := \sqrt{\rho(A^T A)}$

dove $\rho(M) := \max_i |\lambda_i|$ (λ_i : autovalori di M) è il **raggio spettrale** della matrice $M \in \mathbb{R}^{n \times n}$.

Se A è **simmetrica** $\implies \rho(A^T A) = \rho^2(A) \implies \|A\|_2 = \rho(A)$

Teorema. Per una norma verificante la **relazione di compatibilità** si ha $\rho(A) \leq \|A\|$.

Infatti da $\lambda X = AX \implies \|\lambda X\| = \|AX\| \leq \|A\| \cdot \|X\| \implies |\lambda| \leq \|A\|$.

41

Convergenza: condizione necessaria

Tramite la **norma di vettore** si può "misurare" la **lunghezza** del vettore errore di truncamento, cioè la **distanza** tra la soluzione esatta e quella approssimata.

Convergenza: $\lim_{k \rightarrow \infty} \|E^{(k)}\| = 0 \iff \lim_{k \rightarrow \infty} X^{(k)} = \bar{X}$

Teorema. Sia S uno **spazio vettoriale normato** e sia $\Phi: S \rightarrow S$. Se la successione $\{X^{(k)}\} = \{\Phi(X^{(k-1)})\}$ è **convergente** a un valore $\bar{X} \in S$ e l'applicazione Φ è **continua** in $\bar{X} \Rightarrow \bar{X}$ è **punto unito** di Φ , cioè $\bar{X} = \Phi(\bar{X})$.

Dim.

$$\bar{X} = \lim_{k \rightarrow \infty} X^{(k)} = \lim_{k \rightarrow \infty} \Phi(X^{(k-1)}) = \Phi\left(\lim_{k \rightarrow \infty} X^{(k-1)}\right) = \Phi(\bar{X})$$

42

Convergenza: condizione sufficiente

Definizione. Un'applicazione $\Phi: S \rightarrow S$, dove S è uno **spazio normato** è detta **contrazione**, se esiste $\lambda \in (0, 1)$ tale che

$$\|\Phi(X) - \Phi(Y)\| \leq \lambda \|X - Y\| < \|X - Y\| \quad \forall X, Y \in S$$

Teorema. Sia $D \subset \mathbb{R}^n$. Se $\Phi: D \rightarrow D$ è una **contrazione**

\implies • esiste un **unico punto unito** $\bar{X} \in D$ di Φ

• la successione $\{X^{(k)}\} = \{\Phi(X^{(k-1)})\}$ è **convergente** a \bar{X} per ogni **approssimazione iniziale** $X^{(0)} \in D$

43

Contrazione: condizione sufficiente

Matrice Jacobiana di Φ

$$J(X) = \begin{bmatrix} \frac{\partial \varphi_1(X)}{\partial x_1} & \frac{\partial \varphi_1(X)}{\partial x_2} & \dots & \frac{\partial \varphi_1(X)}{\partial x_n} \\ \frac{\partial \varphi_2(X)}{\partial x_1} & \frac{\partial \varphi_2(X)}{\partial x_2} & \dots & \frac{\partial \varphi_2(X)}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial \varphi_n(X)}{\partial x_1} & \frac{\partial \varphi_n(X)}{\partial x_2} & \dots & \frac{\partial \varphi_n(X)}{\partial x_n} \end{bmatrix}$$

Teorema. Se i) le **funzioni di iterazione** $\varphi_1, \varphi_2, \dots, \varphi_n$ sono **continue e parzialmente derivabili** in D ;
ii) esiste $\lambda \in (0, 1)$ tale che $\|J(X)\| \leq \lambda$ per $X \in D$
 $\Rightarrow \Phi$ è una **contrazione** in D

44

Metodi iterativi per sistemi lineari

$$\begin{cases} X^{(k)} = C X^{(k-1)} + Q, & k = 1, 2, \dots & \mathbf{C}: \text{matrice di iterazione} \\ X^{(0)} \in \mathbb{R}^n & \text{dato} \end{cases}$$

$$\Rightarrow x_i^{(k)} = \sum_{j=1}^n c_{ij} x_j^{(k-1)} + q_i, \quad i = 1, \dots, n$$

Un metodo iterativo è detto **convergente** se

$$\lim_{k \rightarrow \infty} \|E^{(k)}\| = 0 \Leftrightarrow \lim_{k \rightarrow \infty} X^{(k)} = \bar{X}$$

Se il metodo iterativo è **convergente**, in assenza di errori di arrotondamento si ottiene la **soluzione esatta** dopo un **numero infinito** di passi.

46

Esempio

La condizione $\|J(X)\| \leq \lambda, X \in D$, è **sicuramente verificata** se

$$\left| \frac{\partial \varphi_i(X)}{\partial x_k} \right| \leq M_{ik} \quad i, k = 1, \dots, n \quad X \in D$$

con

$$\|M\| \leq \lambda < 1 \quad \text{dove } M = [M_{ik}]_{i,k=1}^n$$

Esempio: $n = 2$

$$\begin{cases} f(x, y) = 0 \\ g(x, y) = 0 \end{cases} \Leftrightarrow \begin{cases} x = \varphi(x, y) \\ y = \psi(x, y) \end{cases} \rightarrow \begin{cases} |\varphi_x(X)| \leq M_{11} & |\varphi_y(X)| \leq M_{12} \\ |\psi_x(X)| \leq M_{21} & |\psi_y(X)| \leq M_{22} \end{cases}$$

$$M_{11} + M_{12} \leq \lambda < 1 \quad \text{e} \quad M_{21} + M_{22} \leq \lambda < 1$$

$$\|M\| \leq \lambda < 1 \Leftrightarrow \text{oppure } M_{11} + M_{21} \leq \lambda < 1 \quad \text{e} \quad M_{12} + M_{22} \leq \lambda < 1$$

$$\text{oppure } M_{11}^2 + M_{12}^2 + M_{21}^2 + M_{22}^2 \leq \lambda < 1$$

45

Metodi iterativi per sistemi lineari: condizione sufficiente di convergenza

Matrice Jacobiana di $\Phi = C X + Q$

$$J(X) = \begin{bmatrix} \frac{\partial \varphi_1(X)}{\partial x_1} & \frac{\partial \varphi_1(X)}{\partial x_2} & \dots & \frac{\partial \varphi_1(X)}{\partial x_n} \\ \frac{\partial \varphi_2(X)}{\partial x_1} & \frac{\partial \varphi_2(X)}{\partial x_2} & \dots & \frac{\partial \varphi_2(X)}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial \varphi_n(X)}{\partial x_1} & \frac{\partial \varphi_n(X)}{\partial x_2} & \dots & \frac{\partial \varphi_n(X)}{\partial x_n} \end{bmatrix} = C$$

Corollario. Se esiste $\lambda \in (0, 1)$ tale che $\|C\| \leq \lambda$

$\Rightarrow \Phi$ è una **contrazione** per ogni X

\Rightarrow il metodo iterativo è **convergente**

47

Condizione sufficiente di convergenza

Teorema. Condizione sufficiente affinché un metodo iterativo sia **convergente** a \bar{X} per **qualsiasi scelta** del vettore iniziale $X^{(0)}$, è che

$$\|C\| < 1$$

$$E^{(k)} = \bar{X} - X^{(k)} = (C\bar{X} + Q) - (CX^{(k-1)} + Q) =$$

$$= C(\bar{X} - X^{(k-1)}) = CE^{(k-1)} \Rightarrow E^{(k)} = CE^{(k-1)}$$

$$\|E^{(k)}\| = \|CE^{(k-1)}\| = \|C^2E^{(k-2)}\| = \dots = \|C^k E^{(0)}\| \leq \|C^k\| \cdot \|E^{(0)}\|$$

Relazione di compatibilità

$$\|E^{(k)}\| \leq \|C^k\| \cdot \|E^{(0)}\| = \underbrace{\|C \cdot C \cdot \dots \cdot C\|}_{k \text{ volte}} \cdot \|E^{(0)}\| \leq \|C\|^k \cdot \|E^{(0)}\|$$

$$\text{se } \|C\| < 1 \Rightarrow \lim_{k \rightarrow \infty} \|E^{(k)}\| = 0$$

48

Condizione necessaria e sufficiente di convergenza

Teorema. Una matrice $A \in \mathbb{R}^{n \times n}$ è **convergente** se e solo se $\rho(A) < 1$.

Teorema. Un metodo iterativo **converge** per **qualsiasi scelta** del vettore iniziale $X^{(0)}$, **se e solo se**

$$\rho(C) < 1$$

dove $\rho(C) = \max_{1 \leq i \leq n} |\lambda_i|$ è il **raggio spettrale** della matrice di iterazione C

$$E^{(k)} = CE^{(k-1)} = \dots = C^k E^{(0)}$$

$$\lim_{k \rightarrow \infty} E^{(k)} = \lim_{k \rightarrow \infty} C^k E^{(0)} = 0 \iff \underbrace{\lim_{k \rightarrow \infty} C^k = 0}_{\text{matrice convergente}} \iff \rho(C) < 1$$

49

Criterio d'arresto

Se il metodo iterativo è **convergente**, si arresta il procedimento quando

$$\|X^{(k+1)} - X^{(k)}\| < \varepsilon \quad \varepsilon: \text{ tolleranza prefissata}$$

$$\bullet \|E^{(k)}\| = \|X^{(k)} - \bar{X}\| = \|X^{(k)} - X^{(k+1)} + X^{(k+1)} - \bar{X}\| =$$

$$= \|X^{(k)} - X^{(k+1)} + E^{(k+1)}\| \leq \underbrace{\|X^{(k+1)} - X^{(k)}\|}_{< \varepsilon} + \|E^{(k+1)}\|$$

$$\bullet \|E^{(k+1)}\| \leq \|C\| \cdot \|E^{(k)}\| \leq \|C\| (\|X^{(k+1)} - X^{(k)}\| + \|E^{(k+1)}\|)$$

$$\|E^{(k+1)}\| \leq \frac{\|C\|}{1 - \|C\|} \cdot \|X^{(k+1)} - X^{(k)}\| \leq \frac{\|C\|^{k+1}}{1 - \|C\|} \cdot \|X^{(1)} - X^{(0)}\| \leq \varepsilon$$

Stima a priori: il numero di iterazioni K necessario affinché $\|E^{(K)}\| < \varepsilon$, è dato da

$$K > \log \left(\frac{(1 - \|C\|)\varepsilon}{\|X^{(1)} - X^{(0)}\|} \right) \frac{1}{\log \|C\|}$$

50

Velocità asintotica di convergenza

Se $A \in \mathbb{R}^{n \times n}$ è una **matrice convergente**, vale la proprietà

$$\lim_{k \rightarrow \infty} \sqrt[k]{\|A^k\|} = \rho(A)$$

Dalla relazione

$$\|E^{(k)}\| = \|C^k\| \cdot \|E^{(0)}\|$$

per k "grande" si ottiene

$$\frac{\|E^{(k)}\|}{\|E^{(0)}\|} \leq \|C^k\| \approx \rho^k(C)$$

\Rightarrow L'errore si riduce di un fattore 10^{-m} all'iterazione

$$K \simeq -\frac{m}{\text{Log } \rho(C)}$$

Velocità asintotica di convergenza: $V = -\text{Log } \rho(C)$

51

Esercizio

Data la matrice di iterazione $C(\beta) = \begin{pmatrix} 0 & \beta & -\frac{\beta}{2} \\ 0 & \beta & -\frac{\beta}{2} \\ 0 & -\frac{1}{2} & \frac{1}{4} \end{pmatrix}$, $\beta \in \mathbf{R}$, e il

vettore $Q = (7/8, 7/8, -1/2)^T$,

1.1) determinare per quali valori di β il procedimento iterativo

$$\begin{cases} X^{(k+1)} = C(\beta)X^{(k)} + Q, & k = 0, 1, \dots & X^{(k)} \in \mathbf{R}^3 \\ X^{(0)} \text{ dato} \end{cases}$$

risulta **sicuramente convergente**;

1.2) posto $\beta = 1/2$, $X^{(0)} = (0, 0, 0)^T$, dare una stima del numero di iterazioni necessarie affinché l'approssimazione abbia 5 decimali esatti.

$$\mathbf{1.2)} \quad \beta = \frac{1}{2} \Rightarrow \|C(\frac{1}{2})\|_1 = \frac{3}{2} > 1 \quad \|C(\frac{1}{2})\|_\infty = \frac{3}{4} < 1$$

Attenzione: in **norma uno** non si può stabilire se il metodo converge, si ha invece convergenza in **norma infinito**, infatti

$$\lim_{k \rightarrow \infty} \|E^{(k)}\|_\infty = 0$$

$$K > \log \left(\frac{(1 - \|C(\frac{1}{2})\|_\infty)\varepsilon}{\|X^{(1)} - X^{(0)}\|_\infty} \right) \frac{1}{\log \|C(\frac{1}{2})\|_\infty}$$

$$\|C(\frac{1}{2})\|_\infty = \frac{3}{4}, \quad \|X^{(1)} - X^{(0)}\|_\infty = \|Q\|_\infty = \max |q_i| = \frac{7}{8}, \quad \varepsilon = 0.5 \cdot 10^{-5}$$

$$\Rightarrow K \geq 47$$

Traccia della soluzione

1.1) Condizione sufficiente di convergenza: $\|C\|_1 < 1$ oppure $\|C\|_\infty < 1$

$$\|C\|_1 = \max(2|\beta| + \frac{1}{2}, |\beta| + \frac{1}{4}) = 2|\beta| + \frac{1}{2} \quad \|C\|_1 < 1 \Rightarrow |\beta| < \frac{1}{4}$$

$$\|C\|_\infty = \max(\frac{3}{2}|\beta|, \frac{3}{4}) = \begin{cases} \frac{3}{4} & |\beta| \leq \frac{1}{2} \\ \frac{3}{2}|\beta| & |\beta| > \frac{1}{2} \end{cases} \quad \|C\|_\infty < 1 \Rightarrow |\beta| < \frac{2}{3}$$

Attenzione: le condizioni su β sono diverse a seconda della norma che si sceglie.

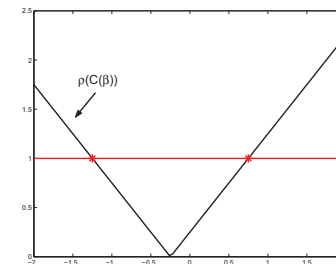
Condizione necessaria e sufficiente di convergenza:

$$\rho(C(\beta)) < 1$$

Autovalori di $C(\beta)$: 0 con molteplicità 2, $\frac{1}{4} + \beta$

Raggio spettrale di $C(\beta)$: $\rho(C(\beta)) = \left| \frac{1}{4} + \beta \right|$

$$\rho(C(\beta)) < 1 \Rightarrow \boxed{-\frac{5}{4} < \beta < \frac{3}{4}}$$



Velocità di convergenza: $V(\beta) = -\text{Log}(\rho(C(\beta))) = -\text{Log} \left| \frac{1}{4} + \beta \right|$

Nota: questo intervallo di β **contiene** entrambi gli intervalli trovati con la condizione sufficiente.

Costruzione di metodi iterativi

$$AX = B \quad \rightarrow \quad X = CX + Q$$

Splitting di A: $A = M + N$ dove M è una matrice **invertibile**.

$$AX = B \rightarrow (M + N)X = B \rightarrow MX = -NX + B \rightarrow$$

$$\rightarrow X = -M^{-1}NX + M^{-1}B \Rightarrow \boxed{C = -M^{-1}N \quad Q = M^{-1}B}$$

Una possibile **decomposizione** di A è

$$\boxed{A = L + D + U}$$

dove $D = \text{diag}(a_{11}, a_{22}, \dots, a_{nn})$ (elementi **diagonali** di A)

$$L = \begin{bmatrix} 0 & 0 & \dots & \dots & 0 \\ a_{21} & 0 & \dots & \dots & 0 \\ a_{31} & a_{32} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{n,n-1} & 0 \end{bmatrix} \quad U = \begin{bmatrix} 0 & a_{12} & a_{13} & \dots & a_{1n} \\ 0 & 0 & a_{23} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 0 & a_{n-1,n} \\ 0 & \dots & \dots & 0 & 0 \end{bmatrix}$$

(elementi di A al di **sotto** della diagonale principale)

(elementi di A al di **sopra** della diagonale principale)

56

Metodo di Jacobi

$$M = D, N = L + U$$

$$\begin{cases} C_J = -D^{-1}(L + U) \\ Q_J = D^{-1}B \end{cases}$$

$$C_J = \begin{bmatrix} 0 & -\frac{a_{12}}{a_{11}} & \dots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & \dots & -\frac{a_{2n}}{a_{22}} \\ \dots & \dots & \dots & \dots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & \dots & 0 \end{bmatrix} \quad Q_J = \begin{bmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ \dots \\ \frac{b_n}{a_{nn}} \end{bmatrix}$$

Algoritmo J

$$X^{(k)} = C_J X^{(k-1)} + Q_J \Rightarrow x_i^{(k+1)} = \frac{1}{a_{ii}} \left(- \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j^{(k)} + b_i \right)$$

$i = 1, 2, \dots, n; k \geq 0$

57

Metodo di Gauss-Seidel

$$M = D + L, N = U \Rightarrow \begin{cases} C_{GS} = -(D + L)^{-1}U \\ Q_{GS} = (D + L)^{-1}B \end{cases}$$

$$X^{(k)} = C_{GS} X^{(k-1)} + Q_{GS}$$

↓

Algoritmo GS

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(- \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} + b_i \right)$$

$i = 1, 2, \dots, n; k \geq 0$

58

Convergenza dei metodi di Jacobi e Gauss-Seidel

- Per **verificare** la convergenza si possono applicare alle matrici di iterazione C_J e C_{GS} le condizioni già viste:

$$\begin{cases} \text{C.S.:} & \|C_J\|_1, \|C_J\|_\infty < 1 \quad \text{e} \quad \|C_{GS}\|_1, \|C_{GS}\|_\infty < 1 \\ \text{C.N.S.:} & \rho(C_J) < 1 \quad \text{e} \quad \rho(C_{GS}) < 1 \end{cases}$$

Attenzione: Le **condizioni di convergenza** per le matrici di iterazione C_J e C_{GS} vanno **verificate** di volta in volta.

- Per alcune matrici A potrebbe convergere **solo uno** dei due metodi.
- Se convergono entrambi i metodi, quello di **Gauss-Seidel** converge **più velocemente**.

59

Convergenza per matrici A con struttura speciale

Matrici diagonalmente dominanti:

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| \quad i = 1, 2, \dots, n \quad |a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ji}| \quad i = 1, 2, \dots, n$$

(diagonale dominante per righe)

(diagonale dominante per colonne)

Condizione sufficiente di convergenza:

Teorema. Se A è **diagonalmente dominante** per righe o per colonne, i **metodi di Jacobi** e **Gauss-Seidel** sono entrambi **convergenti** per qualunque scelta dell'approssimazione iniziale $X^{(0)}$.

Per esempio, se A è **diagonalmente dominante per righe** si ha

$$\|C_J\|_\infty = \max_{1 \leq i \leq n} \left| \sum_{\substack{j=1 \\ j \neq i}}^n -\frac{a_{ij}}{a_{ii}} \right| \leq \max_{1 \leq i \leq n} \left(\sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| \right) \frac{1}{|a_{ii}|} < 1$$

60

Matrici (simmetriche) definite positive:

Una matrice quadrata $A \in \mathbb{R}^{n \times n}$ **simmetrica** è **definita positiva** se

$$X^T A X = \sum_{i,j=1}^n a_{ij} x_i x_j > 0 \quad \forall X \in \mathbb{R}^n$$

Per riconoscere se una matrice è definita positiva si può usare il **criterio di Sylvester**:

Affinché una matrice $A \in \mathbb{R}^{n \times n}$ **simmetrica** sia **definita positiva**, è **necessario e sufficiente** che

$$\det A_k > 0 \quad k = 1, 2, \dots, n,$$

dove A_k sono le **sottomatrici principali di testa** di A .

Condizione sufficiente di convergenza:

Teorema. Se A è (simmetrica) **definita positiva**, il **metodo di Gauss-Seidel** è **convergente** per qualunque scelta dell'approssimazione iniziale $X^{(0)}$.

61

Metodo di rilassamento (S.O.R.)

$$M = \frac{D}{\omega} + L, \quad N = U - \frac{D}{\omega}(1 - \omega) \quad \omega \in \mathbb{R}^+$$

$$\Rightarrow \begin{cases} C_\omega = -\left(\frac{D}{\omega} + L\right)^{-1} \left(U - \frac{D}{\omega}(1 - \omega)\right) \\ Q_\omega = \left(\frac{D}{\omega} + L\right)^{-1} B \end{cases}$$

$$X^{(k)} = C_\omega X^{(k-1)} + Q_\omega = (D + \omega L)^{-1} (-\omega U + (1 - \omega)D) X^{(k)} + \omega (D + \omega L)^{-1} B$$

↓

Algoritmo S.O.R. (successive over relaxation)

$$v_i^{(k)} = \frac{1}{a_{ii}} \left(-\sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} + b_i \right)$$

$$x_i^{(k+1)} = \omega v_i^{(k)} + (1 - \omega) x_i^{(k)}$$

$$i = 1, 2, \dots, n; \quad k \geq 0$$

62

Convergenza del metodo di sovrarilassamento

Condizione necessaria $\left\{ \begin{array}{l} \text{Poiché } \rho(C_\omega) \geq |1 - \omega|, \text{ il } \text{metodo di rilassamento} \\ \text{può } \text{convergere} \text{ solo se } \omega \in (0, 2) \end{array} \right.$

Teorema. Se A è (simmetrica) **definita positiva**, la condizione $\omega \in (0, 2)$ è **necessaria** e **sufficiente** per la convergenza.

Teorema. Se A è (simmetrica) **definita positiva** e **tridiagonale** allora la **velocità di convergenza** è **massima** per

$$\omega = \omega_0 = \frac{2}{1 + \sqrt{1 - \rho(C_{GS})}}$$

63

Esempio 1

Risolvere il problema della **passeggiata casuale** con i **metodi di Jacobi** e **Gauss-Seidel**.

Soluzione

Si tratta di risolvere il sistema lineare

$$\begin{cases} 2p_1 - p_2 & = 1 \\ -p_1 + 2p_2 - p_3 & = 0 \\ \dots & \dots \\ -p_{N-3} + 2p_{N-2} - p_{N-1} & = 0 \\ -p_{N-2} + 2p_{N-1} & = 0 \end{cases}$$

nelle incognite p_i , $i = 1, \dots, N - 1$.

64

Esempio 1: iterazioni

La soluzione esatta è $p_i = 1 - \frac{i}{N}$, $i = 0, \dots, N$.

Per il **metodo di Jacobi** si ha:

N	$\ C_J\ _1$	$\ C_J\ _\infty$	$\rho(C_J)$	$\ X^{(50)} - X^{(49)}\ _\infty$	$\ E^{(50)}\ _\infty$	decimali esatti
11	1	1	0.9595	$0.64 \cdot 10^{-2}$	$0.77 \cdot 10^{-1}$	0
21	1	1	0.9888	$0.95 \cdot 10^{-2}$	0.36	0
51	1	1	0.9981	$0.95 \cdot 10^{-2}$	0.68	0
101	1	1	0.9995	$0.95 \cdot 10^{-2}$	0.81	0

Per il **metodo di Gauss-Seidel** si ha:

N	$\ C_{GS}\ _1$	$\ C_{GS}\ _\infty$	$\rho(C_{GS})$	$\ X^{(50)} - X^{(49)}\ _\infty$	$\ E^{(50)}\ _\infty$	decimali esatti
11	0.9990	0.9980	0.9206	$0.69 \cdot 10^{-3}$	$0.80 \cdot 10^{-2}$	1
21	1	1	0.9778	$0.42 \cdot 10^{-2}$	0.18	0
51	1	1	0.9962	$0.44 \cdot 10^{-2}$	0.55	0
101	1	1	0.9990	$0.44 \cdot 10^{-2}$	0.73	0

Cosa succede se si utilizza il **S.O.R.**? E' possibile determinare ω_{gts} ?

Script MATLAB: metodo di Jacobi

```
% Soluzione di un sistema lineare con metodo di Jacobi
%
% Input:
% A: matrice dei coefficienti
% B: vettore dei termini noti
% epsilon: accuratezza
% Output:
% X: vettore soluzione
% CJ: matrice di iterazione

% Input
A = input('Matrice dei Coefficienti: ');
B = input('Vettore (colonna) dei termini noti: ');

% Calcolo delle dimensioni della matrice
dimA = size(A);
n = dimA(1);
%
% Costruzione matrice di iterazione
%
D = diag(diag(A));
L = tril(A,-1);
U = triu(A,1);
Minv = inv(D);
CJ = -Minv*(L+U);
```

66

```
QJ = Minv*B;
%
% Calcolo autovalori e verifica C.N.S. di convergenza
%
rhoCJ = max(abs(eigs(CJ,1)))
if (rhoCJ >= 1)
    error('Attenzione: ==>> rho > 1')
    return
end
%
% Ciclo iterativo
%
X0 = QJ;
for k=1:50,
    X = CJ*X0+QJ;
    err = norm(X-X0,inf)
    X0 = X;
end,
disp(X')
```

Per costruire la matrice dei coefficienti A e il termine noto B , si possono dare, dalla linea di comando, le istruzioni:

```
>> N=11;A=2*eye(N-1);for i=1:N-1,for j=1:N-1,if abs(i-j)==1,A(i,j)=-1; end,end,end
>> B=zeros(N-1,1);B(1)=1;
```

67

Script MATLAB

```
% Soluzione di un sistema lineare AX=B con l'algoritmo del metodo di Jacobi
%
clear
%
% Input
A = input('Matrice dei Coefficienti: ');
B = input('Vettore (riga) dei termini noti: ');
X0 = input('Vettore (riga) dell'approssimazione iniziale: ');
Kmax = input('Numero massimo di iterazioni: ');
%
% Ciclo iterativo
N = size(A,1);
for k=1:Kmax
    for i=1:N
        X(i) = (-sum(A(i,1:i-1).*X0(1:i-1))-sum(A(i,i+1:N).*X0(i+1:N))+B(i))/A(i,i);
    end
    E(k) = norm(X-X0,inf);
    X0=X;
end
```

Per il **metodo di Gauss-Seidel** si sostituisce l'istruzione del ciclo interno con

```
for i=1:N
    X(i) = (-sum(A(i,1:i-1).*X(1:i-1))-sum(A(i,i+1:N).*X0(i+1:N))+B(i))/A(i,i);
end
```

68

Programma FORTRAN: metodo di Jacobi

```
program jacobi
*
* Soluzione di un sistema lineare con il metodo di Jacobi
*
* Input:
* - n: dimensioni del sistema lineare
* - thres: accuratezza richiesta nell'approssimazione
* - a(n,n): matrice dei coefficienti
* - b(n): vettore dei termini noti
* Output:
* - a(n,n): matrice di iterazione
* - iter-1: numero di iterazioni
* - errmax: differenza tra le ultime due approssimazioni in norma infinito
* - x(n): soluzione approssimata
*
    implicit none
    integer nmax, n, i, j, iter, itermax
    real aux, errmax, thres
*
    parameter (nmax=150)
    real a(nmax,nmax),b(nmax),x(nmax),x0(nmax)
*
* Lettura dei dati di input
* (il vettore dei termini noti viene memorizzato nell'ultima colonna del file)
*
```

69

```
open (20,file='sistlin.dat')
read (20,*) n, thres
if (n .gt. nmax) stop 'n>nmax'
read (20,*) ((a(i,j),j=1,n),b(i),i=1,n)
close(20)
*
* Costruzione della matrice di iterazione (viene memorizzata in a),
* del vettore q (viene memorizzato in b),
* dell'approssimazione iniziale (viene memorizzata in x0)
*
do i = 1, n
    aux = a(i,i)
    b(i) = b(i)/aux
    a(i,i) = 0.
    do j = 1, n+1
        a(i,j) = -a(i,j)/aux
    enddo
    x0(i) = b(i)
enddo
*
* Apertura del file di output
*
open (21,file='output.dat')
write (21,200) ((a(i,j),j=1,n),i=1,n)
200 format (2x,'Matrice di iterazione:'/5(e13.5,2x))
```

70

```
*
* Ciclo iterativo
*
errmax = 100
itermax = 200
iter = 1
do while (errmax.gt.thres.and.iter.le.itermax)
    errmax = 0.
    do i = 1, n
        x(i) = b(i)
        do j = 1, n
            x(i) = x(i) + a(i,j)*x0(j)
        enddo
        errmax = max(errmax,abs(x(i)-x0(i)))
    enddo
    do i = 1, n
        x0(i) = x(i)
    enddo
    iter = iter + 1
enddo
```

71

```

*
* Scrittura dell'output
*
write (21,201) iter-1, errmax, (x(i),i=1,n)
201 format (//,2x,'Numero iterazioni:',i4//,2x,'Errore:',e13.6//,
$      2x,'Vettore Xn: '/6(e13.6,2x))

close(21)

*
stop
end

```

Esercizio

Scrivere un programma (Matlab/Fortran/C) che approssimi la soluzione di un sistema lineare con il metodo del sovrarilassamento. I dati di input (dimensioni del sistema lineare, matrice dei coefficienti, vettore dei termini noti, accuratezza richiesta, parametro di rilassamento) devono essere letti dal file `input.dat`. I dati di output (matrice di iterazione, numero totale di iterazioni, differenza tra due iterazioni successive in norma 1, soluzione approssimata) devono essere scritti sul file `output.dat`.

72

Esercizio

Dato il sistema lineare $AX = B$ dove

$$A = \begin{pmatrix} 1 & 1 & -2 \\ 2 & 1 & 2 \\ 2 & 1 & 1 \end{pmatrix}, \quad B = (1, 0, 0)^T,$$

1.1) verificare quale tra i metodi di Jacobi e Gauss-Seidel risulta convergente;

1.2) approssimare la soluzione del sistema lineare con 6 decimali esatti.

73

Traccia della soluzione

$$\mathbf{1.1)} \quad D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad L = \begin{pmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 2 & 1 & 0 \end{pmatrix} \quad U = \begin{pmatrix} 0 & 1 & -2 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{pmatrix}$$

$$C_J = \begin{pmatrix} 0 & -1 & 2 \\ -2 & 0 & -2 \\ -2 & -1 & 0 \end{pmatrix} \quad C_{GS} = \begin{pmatrix} 0 & -1 & 2 \\ 0 & 2 & -6 \\ 0 & 0 & 2 \end{pmatrix}$$

Entrambe le matrici hanno norma **maggiore di 1**, quindi la **condizione sufficiente** di convergenza **non è soddisfatta**. Per verificare se è soddisfatta la **condizione necessaria e sufficiente** bisogna calcolare il **raggio spettrale** delle matrici di iterazione.

Autovalori di C_J : 0 con molteplicità 3

$\Rightarrow \rho(C_J) = 0 \Rightarrow$ il metodo di Jacobi **converge**

Autovalori di C_{GS} : 0 con molteplicità 1, 2 con molteplicità 2

$\Rightarrow \rho(C_{GS}) = 2 \Rightarrow$ il metodo di Gauss-Seidel **non converge**

74

1.2) Per approssimare la soluzione del sistema lineare con **6 decimali esatti**, si può utilizzare il **metodo di Jacobi** arrestando le iterazioni quando

$$\|E^{(k+1)}\| \simeq \|X^{(k+1)} - X^{(k)}\| \leq 0.5 \cdot 10^{-6} \quad (\text{criterio di arresto})$$

Iterazioni

$$X^{(0)} = Q_J = D^{-1}B = (1, 0, 0)^T$$

$$X^{(1)} = C_J X^{(0)} + Q_J = (1, -2, -2)^T \quad \|X^{(1)} - X^{(0)}\|_\infty = 2$$

$$X^{(2)} = C_J X^{(1)} + Q_J = (1, 2, 0)^T \quad \|X^{(2)} - X^{(1)}\|_\infty = 4$$

$$X^{(3)} = C_J X^{(2)} + Q_J = (1, 2, 0)^T \quad \|X^{(3)} - X^{(2)}\|_\infty = 0$$

Nota 1. In questo caso la soluzione è **esatta**.

Nota 2. La **velocità di convergenza** è $V_J = -\text{Log}(\rho(C_J)) = \infty$.

Cosa si può dire per il metodo S.O.R. ?

75

Esercizio

Dato il sistema lineare $\begin{cases} 5x_1 - x_2 = 9 \\ -x_1 + 5x_2 - 2x_3 = 7 \\ -2x_2 + 5x_3 = 6 \end{cases}$ approssimare

la soluzione con **8 decimali** esatti utilizzando il **metodo SOR**.

Traccia della soluzione

La **matrice dei coefficienti** $A = \begin{bmatrix} 5 & -1 & 0 \\ -1 & 5 & -2 \\ 0 & -2 & 5 \end{bmatrix}$ è **tridiagonale**.

Verifichiamo se è **definita positiva** utilizzando il **criterio di Sylvester**.

$$\det A_1 = 5 > 0$$

$$\det A_2 = \det \begin{bmatrix} 5 & -1 \\ -1 & 5 \end{bmatrix} = 23 > 0 \quad \Rightarrow \quad A \text{ è definita positiva}$$

$$\det A_3 = \det A = 100 > 0$$

Nota. Una matrice **tridiagonale simmetrica** con elementi **positivi** sulla diagonale principale è sempre **definita positiva**

76

Per risolvere il sistema lineare si può utilizzare il **metodo del rilassamento (SOR)** con **parametro ottimo**

$$\omega_0 = \frac{2}{1 + \sqrt{1 - \rho(G_{GS})}}$$

$$C_{GS} = -(D+L)^{-1}U = \begin{bmatrix} 5 & 0 & 0 \\ -1 & 5 & 0 \\ 0 & -2 & 5 \end{bmatrix}^{-1} \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{bmatrix} = \frac{1}{125} \begin{bmatrix} 0 & 25 & 0 \\ 0 & 5 & 50 \\ 0 & 2 & 20 \end{bmatrix}$$

Autovalori di C_{GS} : 0 con molteplicità 2; $\frac{1}{5}$ con molteplicità 1

$$\Rightarrow \rho(C_{GS}) = \frac{1}{5} \Rightarrow \omega_0 \simeq 1.05573$$

Metodo	SOR ω_0	SOR $\omega = 0.8$	Jacobi	Gauss-Seidel
Iterazioni	12	20	25	13
Tempo di calcolo (s)	0.01	0.01	0.03	0.02

77

Condizionamento di un sistema lineare

Il **condizionamento** del problema della soluzione di un sistema lineare è indipendente dal **metodo numerico** scelto per risolverlo.

Il **condizionamento** "misura" quanto una **perturbazione** sui dati di input (matrice dei coefficienti e termine noto) influenzi i risultati (la soluzione).

Un **sistema lineare** si dice **ben condizionato** se a **piccole** variazioni sui dati corrispondono **piccole** variazioni sui risultati.

Viceversa, se a **piccole** variazioni sui dati corrispondono **grandi** variazioni sui risultati, si dice che il sistema è **mal condizionato**.

Quando si approssima la soluzione di un sistema lineare **mal condizionato** bisogna **ridurre** il più possibile gli **errori di arrotondamento**.

78

Errore sul termine noto

Supponiamo che il termine noto B sia affetto da un **errore** δB .

$$AX = B \xrightarrow{\delta B} A(X + \delta X) = B + \delta B$$

Per sottrazione si ricava

$$A \delta X = \delta B \rightarrow \delta X = A^{-1} \delta B$$

Per "misurare" la perturbazione δX indotta su X si ricorre alla norma.

$$\|\delta X\| = \|A^{-1} \delta B\| \leq \|A^{-1}\| \cdot \|\delta B\|$$

$$\|B\| = \|AX\| \leq \|A\| \cdot \|X\|$$

Dividendo termine a termine si trova una maggiorazione per l'**errore relativo** $\|\delta X\|/\|X\|$.

$$\frac{\|\delta X\|}{\|X\|} \leq \underbrace{\|A\| \cdot \|A^{-1}\|}_{K(A)} \frac{\|\delta B\|}{\|B\|} = K(A) \frac{\|\delta B\|}{\|B\|}$$

79

Numero di condizionamento

$K(A) := \|A\| \cdot \|A^{-1}\|$: **numero di condizionamento** della matrice A

$\frac{\|\delta X\|}{\|X\|} \leq K(A) \frac{\|\delta B\|}{\|B\|}$ Rappresenta il **coefficiente di amplificazione** dell'errore relativo sui dati

Si può dimostrare che $1 \leq K(A) \leq +\infty$

Condizionamento **ottimo** (matrici ortogonali) Condizionamento **peggiore** (matrici singolari)

Esempi di matrici malcondizionate: matrici di Hilbert

$$H = \begin{pmatrix} 1 & 1/2 & 1/3 & \dots & 1/n \\ 1/2 & 1/3 & 1/4 & \dots & 1/(n+1) \\ 1/3 & 1/4 & 1/5 & \dots & 1/(n+2) \\ \dots & \dots & \dots & \dots & \dots \\ 1/n & 1/(n+1) & 1/(n+2) & \dots & 1/(2n-1) \end{pmatrix}$$

80

Errore sulla matrice dei coefficienti

Se anche la matrice A è affetta da un **errore** δA si ha

$$\frac{\|\delta X\|}{\|X\|} \leq \frac{K(A)}{1 - \underbrace{K(A) \frac{\|\delta A\|}{\|A\|}}_{\downarrow}} \left(\frac{\|\delta B\|}{\|B\|} + \frac{\|\delta A\|}{\|A\|} \right)$$

Coefficiente di amplificazione

Condizionamento in norma 2

Se A è (simmetrica) **definita positiva** si ha $K_2(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{\lambda_{max}}{\lambda_{min}}$

Infatti $\|A\|_2 = \sqrt{\rho(A^T A)} = \sqrt{\rho(A^2)} = \rho(A) = \lambda_{max}$

$$\|A^{-1}\|_2 = \rho(A^{-1}) = \max_i \frac{1}{\lambda_i} = \frac{1}{\lambda_{min}}$$

81

Esercizio

Data la matrice $A(\lambda) = \begin{pmatrix} 1 & -2 & -\lambda \\ -1 & 0 & -\lambda \\ 0 & 0 & -1 \end{pmatrix}$

- a) individuare i valori di λ per cui $\|A(\lambda)\|_1 \leq 3$;
- b) studiare come varia il numero di condizionamento $K(A(\lambda))$ in norma 1 per $|\lambda| \leq \frac{1}{2}$ e trovarne il massimo;
- c) Dato il sistema $A\left(\frac{1}{2}\right)X = B$, fornire una stima dell'errore relativo $\frac{\|\delta X\|_1}{\|X\|_1}$ corrispondente a un errore relativo $\frac{\|\delta B\|_1}{\|B\|_1} = 10^{-2}$.

82

Traccia della soluzione

a) $\|A(\lambda)\|_1 = \max(2, 2, 2|\lambda|+1) = \begin{cases} 2 & \text{per } 2|\lambda|+1 \leq 2 \Rightarrow |\lambda| \leq \frac{1}{2} \\ 2|\lambda|+1 & \text{per } 2|\lambda|+1 > 2 \Rightarrow |\lambda| > \frac{1}{2} \end{cases}$

$\Rightarrow \|A(\lambda)\|_1 \leq 3$ per $|\lambda| \leq 1$

b) $K(A(\lambda)) = \|A(\lambda)\|_1 \|A^{-1}(\lambda)\|_1$

$A^{-1}(\lambda) = \frac{1}{2} \begin{bmatrix} 0 & -2 & 2\lambda \\ -1 & -1 & 2\lambda \\ 0 & 0 & -2 \end{bmatrix}$ $\|A^{-1}(\lambda)\|_1 = \begin{cases} 3/2 & \text{per } |\lambda| \leq \frac{1}{4} \\ 2|\lambda|+1 & \text{per } |\lambda| > \frac{1}{4} \end{cases}$

$K(A(\lambda)) = \begin{cases} 3 & \text{per } |\lambda| \leq \frac{1}{4} \\ 2(2|\lambda|+1) & \text{per } \frac{1}{4} < |\lambda| \leq \frac{1}{2} \\ (2|\lambda|+1)^2 & \text{per } \frac{1}{2} < |\lambda| \end{cases}$

$\Rightarrow \max_{|\lambda| \leq 1/2} K(A(\lambda)) = K(A(1/2)) = 4$

c) $\frac{\|\delta X\|_1}{\|X\|_1} \leq K(A(1/2)) \frac{\|\delta B\|_1}{\|B\|_1} = 4 \cdot 10^{-2}$

83

Riferimenti bibliografici

L. Gori, *Calcolo Numerico*:

Cap. 2 §§ 2.1-2.5, 2.8-2.11

Cap. 4 §§ 4.1-4.6, 4.8, 4.10 (solo enunciati dei teoremi), 4.12

L. Gori, M.L. Lo Cascio, F. Pitolli, *Esercizi di Calcolo Numerico*:

2.1-2.5, , 2.19-2.25, 2.29, 2.30, 7.15, 7.16, 7.19, 7.35, 7.49, 7.52, 7.57-7.59, 7.64