Università di Pisa

Dipartimento di Informatica
Corso di Laurea Triennale in Informatica

Corso 2° anno - 6 CFU

Calcolo Numerico

Professore:
Prof. Luca Germignani

Autore:
Matteo Giuntoni
Filippo Ghirardini

Anno Accademico 2023/2024

Contents

Calcolo Numerico

Realizzato da: Giuntoni Matteo e Filippo Ghirardini

A.A. 2023-2024

---

1 Aritmetica di Macchina

Per una macchina la scrittura $(x+y)+z$ è diverso da $x+(y+z)$. Vediamo dunque che ci sono alcuni punti focali da considerare per far si che una macchina funzioni correttamente:

- Trovare uno standard per come **memorizzare** i numeri.
- Trovare uno standard per come **manipolare** i numeri.

Da questi due punti possiamo ricondurci ad un solo problema, come andare a **rappresentare** i numeri.

1.1 Teorema di rappresentazione

Teorema 1.1.1. Dato $x \in \mathbb{R}, x \neq 0$[1] e una base di numerazione $B, B \in \mathbb{N}, B > 1$ esistono e sono univocamente determinati:

1. Un intero $p \in \mathbb{Z}$ detto **esponente** della rappresentazione
2. Una successione di numeri naturali $\{d_i\}_{i=1}^{+\infty}$ con $d_i \neq 0, 0 \leq d_i \leq B-1$ e d_i non definitivamente uguali a $B-1$, dette cifre della rappresentazione tali per cui x si scrive in modo **unico** nella seguente forma:
$$x = sign(x)B^p \sum_{i=1}^{+\infty} d_i B^{-1}. \tag{1}$$
dove la sommatoria viene chiamata **mantissa**

Esempio 1.1.1 (Esempio in base 10)**.** Poniamo come numero da rappresentare 0.1 in base 10 è

$$0.1 = +10^0(0.1)$$

Andiamo ora ad analizzare il significato di questo teorema. Esso descrive quella che viene chiamata **rappresentazione in virgola mobile**, in quanto l'esponente p on è determinato in modo da avere la parte intera nulla. Le cose da considerare in questo teorema sono:

- La condizione $d_i \neq 0$ e d_i non definitivamente uguale a $B-1$ sono introdotte per garantire l'unicità delle rappresentazioni. Ad esempio:

  $$B = 10 \text{ abbiamo } 1 = +10^1(1 \cdot 10^{-1}) = +10^2(0 \cdot 10^{-1} + 1 \cdot 10^{-1})$$

  Quindi due rappresentazioni diverse per lo stesso numero, però considerando le condizioni scritte sopra la seconda non risulta accettabile perché la prima cifra è nulla.
  Questa clausola ci garantisce anche l'unicità delle rappresentazioni nei numeri **periodici**:

  $$0.\overline{9} = 10^0(0.99\ldots9)$$

  Non è ammissibile in quanto è definitivamente uguale a $B-1$.

- Questa rappresentazione si estende anche all'insieme dei numeri complessi del tipo $z = a + ib$, utilizzando una rappresentazione come coppie di numeri reali del tipo (a, b).

Possiamo dedurre che visto che stiamo lavorano con registri di memoria di un calcolatore con memoria a numero finito, anche la quantità di cifre rappresentabili saranno a numero finito esso viene chiamato **insieme dei numeri di macchina**.

Dal teorema di rappresentazione in base di un numero reale può avvenire assegnando delle posizioni di memoria per il segno, per l'esponente e per le cifre della rappresentazione.

[1] Lo zero viene utilizzato dalla macchina per alcune operazioni come il confronto, quindi deve averlo ma lo rappresenta in un modo particolare

Definizione 1.1.1 (Insieme dei numeri di macchina)**.** *Si definisce l'insieme dei numeri di macchina in rappresentazione floating point con t cifre, base B e range $-m, M$ l'insieme dei numeri reali.*

$$\mathbb{F}(B,t,m,M) = \{0\} \cup \{s \in \mathbb{R} : x = \pm B^p \sum_{i=1}^{t} d_i B^{-1}, d_1 \neq 0, -m \leq p \leq M\}$$

Si osserva in questa definizione che:

- L'insieme $\mathbb{F}$ ha cardinalità **finita**: $N = 2B^{t-1}(B-1)(M+m+1)+1$.
- L'insieme dei numeri di macchina $\mathbb{F}(B,t,m,M)$ è **simmetrico** rispetto all'origine.
- Possiamo definire $\Omega = B^M \sum_{i=1}^{t}(B-1)b^{-1}$ come il **più grande** numero macchina e $\omega = +B^{-m}B^{-1}$ come invece il **più piccolo** positivo.
- Posto un $x = B^P \sum_{i=1}^{t} d_i B^{-1}$ possiamo definire il suo **successivo** numero di macchina come $y = B^p(\sum_{i=1}^{t-1} d_i B^{-1} + (d_t+1)B^{-t})$.
  Da qui vediamo che la distanza $y - x = B^p - t$ porta i numeri ad essere **non equidistanti** fra di loro, quindi la distanza aumenta con l'avvicinarci a Ω.
  Questo ci fa comodo perché ci interessa l'**errore relativo**, quindi su numeri piccoli ci serve un errore piccolo mentre su numeri grandi posso fare errori grandi.

Esempio 1.1.2. Facciamo ora un esempio in cui andiamo a rappresentare il numero successivo di $x = B^p \sum_{t}^{i=1} d_i B^{-1}$. Esso si può scrivere come $y = B^p\left(\sum_{i=1}^{t-1} d_i B^{-1} + (d_t+1)B^{-t}\right)$.
Mentre si può scrivere la distanza fra questi due valori come $y - x = B^p - t$.

E' stato fissato uno standard IEEE 754 negli anni 70/80 che dice che, visto ci sono macchine che hanno metodi di rappresentazione diversi, bisogna fissare un standard, ovvero $B = 2$ con registri a 32 o 64 bit.

Questa rappresentazione ha uno svantaggio che può sembrare minimo ma non lo è, lo 0 si rappresenta due volte con $-0, +0$. Per ovviare a questo problema si è andato ad abbandonare questa rappresentazione in esponenti ma si rappresentato i numeri nel seguente modo: $p_1 2^0 + p_1 2^1 + \cdots + p_1 1 s^1 0$ che rappresentano numeri da 0 a $2^1 1 - 1$ quindi 2047 numeri, mentre lo 0 si può scrivere come:

- O tenendo tutti i valori a 0
- Oppure tendendo tutti i valori a 1

In entrambi i casi abbiamo un range di valori che va da $[-1022, 1024]$. A questo punto ho 2^{P-1022} numeri che la macchina rappresenta come $\pm 2^{P-1022}(0.1d_1 \dots d_{52})$.
Impostando questo standard abbiamo $\Omega = 2^{1024}(01\dots1)_2$ e $\omega = 2^{-1022}(101)_2$.

Osservazione 1.1.1. Quando $p = 0$ abbiamo i numeri che si trovano nella porzione della retta dei numeri che è compresa fra $-\omega$ ω e possiamo qui avere anche tutti 0 e quindi si introduce il caso dei numeri denormalizzati.

Se abbiamo l'esponente uguale a tutti 1, la convenzione è che tutte le cifre della mantissa sono tutti uguali a 0/1 questo numero indica il $\pm\infty$ altrimenti sta a significare NaN (not a number). Questi valori ci permettono di gestire forme indeterminate.

1.2 Errore di rappresentazione

Quando si va a rappresentare un numero reale non nullo $x \in \mathbb{R}$ e con $x \neq 0$ si può andare a commettere degli errori di rappresentazione detto anche **errore relativo di approssimazione**, e di definisce come, prendendo un $\tilde{x} \in \mathbb{F}(B,t,m,M)$

$$\epsilon_x = \frac{\tilde{x} - x}{x} = \frac{\eta x}{x}, x \neq 0$$

Definiamo $|\epsilon_x| = \left|\frac{\tilde{x}-x}{x}\right| \leq \frac{B^{P-t}}{|x|} \leq \frac{B^{P-t}}{B^{P-1}} = B^{1-t} = u$ la u è definita come **precisione di macchina**.

Andiamo inoltre a definire le condizioni di underflow e overflow. Dato un $x \in \mathbb{R}, x \neq 0$ abbiamo che:

1. Se $|x| < \omega$ o $|x| > \Omega$ overflow. In questo caso si va ad associare il $+\infty$.
2. Se invece $\omega \leq |x| \leq \Omega$ abbiamo underflow. In questo caso allora prendiamo una $x = B^p \sum_{i=1}^{\infty} d_i B^{-1} \longrightarrow B^p \sum_{i=1}^{t} d_i B^{-1} = \tilde{x}$ che è una approssimazione

1.3 Operazioni di macchina

Consideriamo ora due numeri $x, y \in \mathbb{F}$ e chiediamoci perché la macchina non possa fare l'operazione $x + y$. La risposta è che i risultati da questa operazione di ritornano fra i numeri di macchina. Per ovviare a questo problema dovremo usare le Operazioni di macchina che si identificano come $\oplus \ominus \otimes \oslash$. Nel nostro caso l'addizione di macchina $x \oplus y = troncamento(x+y) = (x+y)(1+\epsilon_1)$ con $|\epsilon_1| \leq u$ con e_1 detto errore locale dell'operazione.

Esempio 1.3.1. Supponiamo di dover calcolare in macchina la funzione $f(x) = \frac{x-1}{x}$. In macchina questa funzione corrisponderebbe a $g(\tilde{x}) = (\tilde{x} \ominus 1) \oslash \tilde{x}$. Abbiamo quindi:

$$g(\tilde{x}) = \frac{(x(1+\epsilon_x)-1)(1+\epsilon_1)}{x(1+\epsilon_x)} \cdot (1+\epsilon_1) = \frac{(x(1+\epsilon_x)-1)(1+\epsilon_1+\epsilon_2)}{x(1+\epsilon_x)} = \frac{(x(1+\epsilon_x)-1)(1+\epsilon_1+\epsilon_2-\epsilon_x)}{x}$$

$$g(\tilde{x}) = (\tilde{x} \oplus 1) \oslash \tilde{x} = \frac{(x(1+\epsilon_x)-1)(1+\epsilon_1+\epsilon_2-\epsilon_x)}{x}$$

$$\frac{g(\tilde{x}-f(x))}{f(x)} = \frac{((x-1)/x) + (\epsilon_1+\epsilon_2)((x-1)/x) + \epsilon_2/x - ((x-1)/x)}{(x-1)/x} = \epsilon_1 + \epsilon_2 - \frac{\epsilon_x}{x-1}$$

Esempio 1.3.2. Supponiamo ora di calcolare la funzione $f(x) = \frac{x-1}{x}$ in un altro modo, $g_2(\tilde{x}) = \frac{g_2(\tilde{x})-f(x)}{f(x)}$ ed andiamo a fare l'analisi dell'errore.

$$\frac{g_1(\tilde{x}) - f(x)}{f(x)} \doteq \epsilon_1 + \epsilon_2 + \frac{\epsilon_1}{(x-1)} \text{ Questo è il risultato di un analisi al primo ordine}$$

$$g_2(\tilde{x}) = 1 \ominus \frac{1}{\tilde{x}}(1+\delta_1) = 1 \ominus \frac{1}{x}(1+\delta_1)(1-\epsilon_x) = [1 - \frac{1}{x}(1-\delta_1)(1-\epsilon_1)](1+\delta_2)$$

$$\doteq (1+\delta_1) - \frac{1}{x}(1+\delta_1+\delta_2+\epsilon_x) \doteq (1-\frac{1}{x}) + \delta_2(1-\frac{1}{x}) - \frac{\delta_1}{x} + \frac{\epsilon_x}{x} \doteq \delta_2 - \frac{\delta_1}{x-1} + \frac{\epsilon_x}{x-1}$$

$$\frac{g_2(\tilde{x}) - f(x)}{f(x)} = \delta_2 - \frac{\delta_1}{x-1} + \frac{\epsilon_x}{x-1}$$

Questo è il risultato finale dove $\delta_2 - \frac{\delta_1}{x-1}$ viene definita come parte stabilità mentre $\delta_2 - \frac{\delta_1}{x-1}$ viene chiamato condizionamento,il risultato finale viene definito invece numero stabile.

2 Calcolo degli errori

Supponiamo di avere una funzione $f : [a, b] \to \mathbb{R}$ e $f \neq 0$, per andare a calcolare questa funzione come già visto usiamo un algoritmo che esprime tale valore come risutlato di una sequanza di operazioni aritmentiche. Questa rappresentazione come abbiamo già potuto verificare con esempi produce degli errori di approssimazione. Questi errori possono essere suddivisini in 3 tipologie.

Definizione 2.0.1 (Errore inerente o inevitabile)**.** *Si dice errore inerente o inevitabile generato nel calcolo di $f(x) \neq 0$ la quantità:*

$$\epsilon_{in} = \frac{f(\tilde{x}) - f(x)}{f(x)}$$

Definizione 2.0.2 (Errore algoritmico)**.** *Si dice errore algoritmico generato nel calcolo di $f(x) \neq 0$ la quantità:*

$$\epsilon_{alg} = \frac{g(\tilde{x}) - f(\tilde{x})}{f(\tilde{x})}$$

Definizione 2.0.3 (Errore totale)**.** *Si dice errore algoritmico totale nel calcolo di $f(x) \neq 0$ mediante l'algoritmo specificato da g la quantità:*

$$\epsilon_{tot} = \frac{g(\tilde{x}) - f(x)}{f(x)}$$

Osservazione 2.0.1. Vediamo che se $|\epsilon_{in}|$ è grande il problema si definisce **problema mal condizionato**. Mentre se $|\epsilon_{alg}$ è grande l'algoritmo si dice che **algoritmo è numericamente instabile**.

3 Matrici

3.1 Norme vettoriali

Sono uno strumento che ci permette di definire una distanza tra due vettori.

Definizione 3.1.1 (Norma vettoriale)**.** *È una funzione del tipo* $f : \mathbb{R}^n \to \mathbb{R}$ *che deve soddisfare tre proprietà:*

1. $f(x) \geq 0 \wedge f(x) = 0 \Leftrightarrow x = 0$
2. $f(\alpha x) = |\alpha| f(x) \quad \forall \alpha \in \mathbb{R} \quad \forall x \in \mathbb{R}^n$
3. ***Disuguaglianza triangolare****:* $f(x + y) \leq f(x) + f(y) \quad \forall x, y \in \mathbb{R}^n$

e la indichiamo come

$$f(x) = \|x\|$$

Detto questo possiamo definire una distanza come:

$$dist(x, y) = \|x - y\| \tag{2}$$

Le tre proprietà ci danno alcune informazioni sulla distanza:

1. La distanza deve essere non negativa e valere 0 solo se i due vettori coincidono
2. La distanza tra x e y deve essere uguale a quella tra y e x
3. $\|x - y\| = \|(x - a) + (a - y)\| \leq \|x - a\| + \|a - y\|$

Definizione 3.1.2 (Distanza euclidea - Norma 2)**.**

$$f(x) = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2} > \|x\|_2 \tag{3}$$

Definizione 3.1.3 (Norma infinito)**.**

$$f(x) = max|x_1| = \|x\|_\infty \tag{4}$$

Definizione 3.1.4 (Norma 1)**.**

$$f(x) = \sum_{i=1}^{n} |x_i| = \|x\|_1 \tag{5}$$

Esempio 3.1.1. Prendiamo due vettori

$$\begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \quad \begin{Bmatrix} 2 \\ -2 \end{Bmatrix}$$

e calcoliamo le varie norme:

$$\|x\|_2 = \sqrt{2} \quad \|x\|_\infty = 1 \quad \|x\|_1 = 2$$

Definizione 3.1.5 (Norma matriciale)**.** *È una funzione del tipo* $f : \mathbb{R}^{n \times n} \to \mathbb{R}$ *che deve soddisfare tre proprietà:*

1. $f(A) \geq 0 \wedge f(A) = 0 \Leftrightarrow A = 0$
2. $f(\alpha A) = |\alpha| f(x) \quad \forall \alpha \in \mathbb{R} \quad \forall A \in \mathbb{R}^{n \times n}$
3. ***Disuguaglianza triangolare****:* $f(A + B) \leq f(A) + f(B) \quad \forall A, B \in \mathbb{R}^{n \times n}$
4. $f(A \cdot B) \leq f(A) f(B)$

e la indichiamo come

$$f(A) = \|A\|$$

4 Condizionamento

Studiare il condizionamento di un problema in forma $Ax = b$ significa chiedersi di quanto cambia la soluzione perturbando di poco A e B.

4.1 Metodi diretti

Dato un sistema lineare $Ax = b$ le soluzioni possono essere trovate tramite

$$x_i = f_i(a_{11}, \dots, a_{1n}, b1, \dots, b_n) \quad i : 1 \dots n$$

Si può partire studiando la forma di A per cercare dei sistemi risolvibili facilmente.

4.1.1 Matrice diagonale

Il primo caso è quando A è una matrice **diagonale**:

$$a_{ij} = 0 \Longleftarrow i \neq j$$

Il determinante di una matrice diagonale è $det(A) = \prod_{i=1}^{n} a_i$ e il determinante è diverso da 0 solo se tutti gli elementi della diagonale lo sono. Possiamo riscrivere la matrice in un sistema di equazioni lineari:

$$Ax = b \Leftrightarrow \begin{cases} a_1 x_1 = b_1 \Leftrightarrow x1 = \frac{b1}{a1} \\ \vdots \\ a_n x_n = b_n \Leftrightarrow x_n = \frac{b_n}{a_n} \end{cases}$$

Questo sistema lo risolviamo in $O(n)$. Dato però che ridurre una matrice normale in una diagonale è un processo complesso, non conviene farlo.

4.1.2 Matrice triangolare

Una matrice è **triangolare inferiore** quando $a_{ij} = 0$ per $j > i$, mentre è **triangolare superiore** quando $a_{ij} = 0$ per $i > j$.
Per calcolare il determinante di una matrice triangolare superiore con Laplace facciamo:

$$A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ \vdots & \ddots & 0 & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_{n1} & \cdots & \cdots & a_{nn} \end{bmatrix}$$

Per risolvere $Ax = b$ con una matrice triangolare, vediamo che il sistema associato ci porta ad avere prima un equazione con tutte le incognite fino all'ultima con una sola incognita. È intuitivo che per calcolarne la soluzione è sufficiente partire dall'ultima ed eseguire una **sostituzione all'indietro**.

```
function[x]=backward_substitution(a,b)
   n=length(b);
   x = zeros(n,1);
   for k=n:-1:1
      s=0;
      for j=k+1:n
         s=s+a(k,j)*x(j);
      end
      x(k)=(b(k)-s)/a(k,k);
   end
```

Dato che l'operazione moltiplicativa in macchina richiede leggermente più lunga di quella della somma, è sufficiente considerare le prime per calcolare la complessità di questo programma.
La complessità al caso pessimo è quindi $\frac{n \cdot (n+1)}{2} = O(n^2) = \frac{n^2}{2} + O(n)$.

Esempio 4.1.1 (Matrice bidiagonale superiore)**.** Supponiamo di avere la seguente matrice bidiagonale superiore:

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & a_{n-1n} \\ 0 & 0 & 0 & a_{nn} \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & b_{n-1} \\ 0 & 0 & 0 & a_n \end{bmatrix}$$

In questo caso il codice si può cambiare sostituendo il secondo ciclo con l'operazione

```
s=a(k,k+1)*x(k+1);
```

e la complessità diventa $O(n)$.

5 Metodi iterativi

Una classe di metodi che permette di risolvere sistemi lineari è quella dei metodi iterativi. L'idea è di costruire una successione di vettori partendo dalla matrice che convergano alla soluzione del sistema lineare.

$$\{X_k\}_{k\in\mathbb{N}}$$

Osservazione 5.0.1. Non potendo generare infiniti termini, ad un certo punto ci sarà bisogno di fermarsi quando si pensa di essere sufficientemente vicini alla soluzione.

Diciamo che

$$\lim_{k\to+\infty} x_k \Leftrightarrow \lim_{k\to+\infty} ||x_k - x||_\infty = 0 \tag{6}$$

Questo è vero perché:

$$\forall j = 1,\dots,n \quad 0 \le |x_j^{(k)} - x_j| \le ||x^{(k)} - x||_\infty$$

Abbiamo una matrice invertibile A. Supponiamo di scomporre la matrice come:

$$A = M - N \tag{7}$$

con l'assunzione che anche M sia invertibile ($det(M) = 0$). A questo punto possiamo dire che:

$$Ax = b \Leftrightarrow (M-N)x = b \Leftrightarrow Mx = Nx + b \Leftrightarrow x = M^{-1}Nx + M{-}1b \Leftrightarrow x = Px + q$$

Dovendo trovare la soluzione di $x = Px + q$, possiamo scegliere un vettore iniziale $x_0 \in \mathbb{R}^n$ e costruirci la successione di vettori

$$x_{k+1} = Px_k + q \tag{8}$$

Se questa successione converge ho trovato la soluzione del sistema lineare.

Osservazione 5.0.2. Nell'implementazione pratica non userò mai l'equazione 8 ma invece

$$Mx_{k+1} = Nx_k + b \tag{9}$$

Esempio 5.0.1. Prendiamo

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad b = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad Ax = b \Leftrightarrow x = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Partiamo definendo le due matrici per la scomposizione:

$$M = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad N = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

e calcolando la matrice P

$$P = M^{-1}N = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{2} \\ -\frac{1}{2} & 0 \end{bmatrix}$$

e il vettore q Iniziamo il metodo iterativo:

$$x^{k+1} = Px^{(k)} = \begin{bmatrix} 0 & -\frac{1}{2} \\ -\frac{1}{2} & 0 \end{bmatrix} x^{(k)}$$
$$x^{(1)} = \begin{bmatrix} 0 & -\frac{1}{2} \\ -\frac{1}{2} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}y \\ -\frac{1}{2}x \end{bmatrix}$$
$$x^{(2)} = \begin{bmatrix} 0 & -\frac{1}{2} \\ -\frac{1}{2} & 0 \end{bmatrix} \begin{bmatrix} -\frac{1}{2}y \\ -\frac{1}{2}x \end{bmatrix} = \begin{bmatrix} \frac{1}{4}x \\ \frac{1}{4}y \end{bmatrix}$$

e notiamo che la successione tende a

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Se prendiamo invece

$$M = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad N = \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix}$$

con

$$P = M^{-1}N = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix} = \begin{bmatrix} 0 & -2 & -2 & 0 \end{bmatrix}$$

abbiamo che la successione **diverge**.

Definizione 5.0.1. *Un metodo iterativo è convergente se*

$$\forall x^{(0)} \in \mathbb{R}^n \quad x_k \to x \tag{10}$$

Ovvero se per ogni vettore di partenza scelto, il metodo converge.

Teorema 5.0.1. Dato $x^{(k+1)} = Px^{(k)} + q$, se

$$\exists ||\cdot|| \text{ tc } ||P|| < 1 \tag{11}$$

allora il metodo è convergente.

Esempio 5.0.2. Data una matrice

$$A = \begin{bmatrix} 3 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 3 \end{bmatrix}$$

definiamo le matrici per la scomposizione

$$M = \begin{bmatrix} 3 & & \\ & \ddots & \\ & & 3 \end{bmatrix} \quad N = \begin{bmatrix} 0 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 0 \end{bmatrix}$$

Per capire se il metodo è convergente dobbiamo calcolare la norma infinito di P:

$$M^{-1}N = \begin{bmatrix} 0 & \frac{1}{3} & & \\ \frac{1}{3} & \ddots & \ddots & \\ & \ddots & \ddots & \frac{1}{3} \\ & & \frac{1}{3} & 0 \end{bmatrix} \quad ||P||_\infty = \frac{1}{3} + \frac{1}{3} = \frac{2}{3} < 1$$

Il problema di questo metodo iterativo è che ha complessità $O(n)$ per ogni iterazione e non è quindi competitivo con l'eliminazione Gaussiana.

Esempio 5.0.3. Data una matrice

$$A = \begin{bmatrix} T & I & & \\ -I & \ddots & \ddots & \\ & \ddots & \ddots & I \\ & & -I & T \end{bmatrix} \quad T = \begin{bmatrix} 5 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 5 \end{bmatrix}$$

In questo caso i metodi iterativi sono vantaggiosi poiché anche se la matrice è predominante diagonale (e quindi permette Gauss senza problemi), a causa dell'effetto del fill-in lo rende sconveniente.

Esempio 5.0.4. Data una matrice

$$A = \begin{bmatrix} n & -1 & \dots & -1 \\ -1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -1 \\ -1 & \dots & -1 & n \end{bmatrix}$$

calcoliamo P

$$N = \begin{bmatrix} 0 & 1 & \dots & 1 \\ 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 1 & \dots & 1 & 0 \end{bmatrix} \quad P = \begin{bmatrix} 0 & \frac{1}{n} & \dots & \frac{1}{n} \\ \frac{1}{n} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{1}{n} \\ \frac{1}{n} & \dots & \frac{1}{n} & 0 \end{bmatrix}$$

La sua norma vale

$$||P||_\infty = \frac{n-1}{n}$$

e quindi converge.

Note 5.0.1. Se la norma vale 1 non posso dire nulla sulla convergenza.

Teorema 5.0.2.

$$x^{(k+1)} = Px^{(k)} + q \text{ è convergente } \Leftrightarrow \phi(P) < 1 \tag{12}$$

Esempio 5.0.5. Data una matrice

$$A = \begin{bmatrix} 1 & & & \alpha \\ -1 & \ddots & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}$$

troviamo le matrici per la scomposizione

$$M = I \quad N = \begin{bmatrix} 0 & & & -\alpha \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix}$$

Per quali α il metodo è convergente?
Troviamo la fattorizzazione LU della matrice per il calcolo del determinante e otteniamo così il polinomio caratteristico:

$$x^n + \alpha$$

Dobbiamo poi trovare il modulo degli autovalori per poterli confrontare:

$$x^n = -\alpha$$
$$|\lambda|^n = |-\alpha| \Rightarrow |\lambda| = \sqrt[n]{|\alpha|}$$
$$\sqrt[n]{|\alpha|} < 1 \Leftrightarrow |\alpha| < 1$$