

Analisi di modelli e
Algoritmi per reti
Multiplex

Garau Andrea

Matematica Computazionale

Analisi di modelli e
Algoritmi per reti
Multipliex



Garau Andrea

47017



Relatore: Prof. *Giuseppe Rodriguez*
Co-relatatrice: *Fenu Caterina*
Controrelatore: *Fraschini Matteo*
Studente: *Garau Andrea*

Copyright © *tutti i diritti sono riservati*

Questo documento è stato composto dall'autore mediante L^AT_EX, su un sistema GNU OS: Arch Linux x86_64 con Kernel Release 4.10.2 – 1 – ARCH. Per la stesione del documento, così come per il codice prodotto, si è usato il programma free software Emacs . L'unico software commerciale utilizzato, per i soli esperimenti numerici, è Matlab . I grafici sono stati prodotti con il pacchetto TikZ.

Per maggiori informazioni in merito garau.and@gmail.com

Ringraziamenti

Aha... le care e solite formalità¹, sento il dovere di fare dei ringraziamenti². La prima persona che devo certamente ringraziare, e **non** è una formalità, è il Prof. Rodriguez, solo un *debianizzato* poteva tollerare la scelta che ho fatto senza nessuna contrariazione.

Mi sento di dover ringraziare anche altre persone. Prima di tutto la mia compagna, perché senza di Lei le mie scelte non avrebbero avuto molto senso. Alle persone che non ci sono più, perché il loro non esserci ha permesso a me di essere qui.

Poche persone al mondo, praticamente nessuna, può dire di aver lavorato e lavora per un mondo “libero”, *Richard Stallman* è una di queste. Non smetterò mai di stupirmi e studiare le potenzialità di **Emacs**, che da *semplice* editor è diventato un Sistema Operativo con un suo *personale linguaggio*: **Elisp** un dialetto che discende direttamente dal *Linguaggio Universale*: **Lisp** il padre di tutti i linguaggi³. Per tutto questo lavoro è stato usato Emacs, sia per L^AT_EX, e i suoi tricks, che per gli algoritmi svolti in Matlab⁴.

Prima di abbandonare i Ringraziamenti devo assolutamente *ringraziare* una persona, o meglio *la* persona che ha reso possibile tutto questo.

Quella che è rimasta sempre al mio fianco sia nei momenti belli che brutti, facili e difficili, quella che con me ha condiviso davvero tutto, che non mi ha *mai* scoraggiato, che mi ha convinto che la *conoscenza* non può esistere senza la *curiosità*.

Questa persona mi ha permesso di fare quello che altre persone⁵ non ha neanche il coraggio di pensare. Voglio ringraziare me stesso! Grazie Andrea per avere sopportato tutto questo⁶.

¹Il lettore frettoloso può saltare questa parte senza compromettere nulla sulla comprensione del lavoro.

²Quello superfrettoloso può direttamente saltare su un altro lavoro.

³È un anno più giovane di FORTRAN.

⁴Unica nota proprietaria del lavoro.

⁵Praticamente l'intera popolazione degli Ingegneri.

⁶...prego!

Introduzione

LOBIETTIVO di questo elaborato è quello di analizzare alcuni algoritmi numerici, e simularli praticamente. Non c'è nulla che la matematica non possa descrivere. Conoscere il **modello matematico** del fenomeno, un complesso di formule che descrivono il suo comportamento, vuol dire conoscere **tutto** del fenomeno. Talvolta, diciamo pure il 99.99...9% della volte, le equazioni così ottenute sono troppo complesse per essere risolte, si associa allora un **problema numerico** allo scopo di renderlo risolubile numericamente su un calcolatore, attraverso una sequenza: **l'algoritmo**.

Definizione 0.0.1: Algoritmo

Una sequenza univoca di un numero finito di operazioni elementari che stabilisce come calcolare la soluzione di un problema, assegnati certi dati iniziali.

Un concetto **fondamentale** è quello di **problema ben posto**. J.Hadamard né diede una definizione.

Definizione 0.0.2: Problema ben posto

Un problema è **ben posto** se esso possiede, in un prefissato **campo di definizione, una e una sola** soluzione e questa dipende con **continuità** dai dati: In caso contrario, viene detto mal posto.

Organizzazione del lavoro

Capitolo 1. Nel primo capitolo si esamineranno i concetti matematici che serviranno nel resto del lavoro. Si richiamano alcune definizioni come spazi, rango etc., si richiamano le definizioni di matrice e alcune strutture particolari. Nell'ultima parte si analizzano gli algoritmi di Lanczos e si vedono le formule di quadratura.

Capitolo 2. Si descrivono i concetti generli di teoria delle reti. Si danno le definizioni di nodo e arco di rete in generale, si introduce la matrice di adiacenza e di incidenza. Si introducono i cammini e i percorsi di una rete.

Capitolo 3. Vengono analizzate delle reti più complesse: le reti *multiplex*, si danno delle definizioni e le si descrivono.

Capitolo 4 L'implementazione e le prove numeriche delle simulazioni eseguite. Inizialmente si analizzano esempi random di reti e di seguito vengono prese tre tipi di reti prese su dati reali:

- ① rete yeast (2114 nodi e 4480 archi): che descrive l'interazione proteica del lievito;
- ② rete power (4941 nodi e 13188 archi): che descrive la rete elettrica degli stati dell'ovest degli Stati Uniti;
- ③ la rete facebook di New Orleans (63731 nodi e 1545686 archi).

I dati vengono riassunti su dei grafici.

We are all born mad. Some remain so.
Samuel Beckett *Waiting for Godot*

Indice

1 | Preliminari

- 1.1 Spazi 1
- 1.2 Matrici 4
- 1.2.1 Autovalori e autovettori 5
- 1.3 Matrici di forma particolare 7
- 1.3.1 Hessenberg 7
- 1.4 Ortogonalizzazione di Gram-Schmidt 12
- 1.5 Polinomi ortogonali 14
- 1.6 Iterazioni in sottospazi di Krylov 17
- 1.6.1 L'iterazione di Arnoldi 18
- 1.6.2 L'iterazione di Lanczos 20
- 1.7 Funzioni matriciali o di matrice 22
- 1.7.1 Regola di Gauss 24
- 1.7.2 La regola di Gauss-Radau 24
- 1.7.3 Forme bilineari 25

2 | Concetti generali di teoria delle reti

- 2.1 Reti e matrici 29
- 2.2 Cammini e percorsi 30

3 | Multilayer Network

- 3.0.1 Classificazione reti multistrato 33
- 3.1 Rappresentazione tensoriale 34
- 3.2 *Appiattimento* del tensore 34
- 3.3 Rappresentazione di supra-adjacency 35
- 3.4 Reti multiple e reti multi relazionali 35

4

Implementazione e prove numeriche

- 4.1 Risultati ottenuti 42
- 4.1.1 Prove un pò più complesse 44
- 4.1.2 Prove *reali* 48
- Riferimenti bibliografici 54

Elenco delle figure

3.1	<i>Un esempio generale di multilayer network, $M = (V_M, E_M, V, L)$. Sono presenti quattro nodi $V = \{1, 2, 3, 4\}$, e due aspetti, ciascuno ha due strati elementari, dunque la combinazione di questi strati elementari, porta all'insieme dei layer (strati). Ci sono un totale di 4 differenti strati. Ciascuno contiene alcuni nodi di V. Le connessioni blu (linee blu) rappresentano gli archi intra-layer, quelle verdi gli archi inter-layer. A destra è riportato il grafo $G_M = (V_M, E_M)$ della stessa multilayer network.</i>	32
3.2	<i>Grafo intra-strato, inter-strato, accoppiato, della multilayer network della figura 3.1, è stato usato un colore arancione per enfatizzare il grafo accoppiato.</i>	33
3.3	<i>multilayer network, si sono presi archi indirezionali. Ci sono due aspetti. Sono stati usati colori differenti per rappresentare gli archi intra-strato (blu), e inter-strato (verde). Tutti gli archi sono archi accoppiati perché i nodi sono adiacenti soltanto a se stessi, e a nessun altro nodo. Gli archi di accoppiamento nel primo aspetto sono categorici, perché esiste un arco tra i nodi in ciascuno strato, mentre nel secondo aspetto è ordinale, infatti non tutti i nodi sono connessi per differenti strati.</i>	37
4.1	<i>Tempo di calcolo per rete sprand(10), con 10 nodi, applicando l'algoritmo iterativo Lanczos/Quadratura, confronto con l'approssimazione di Padé $\text{expm}(\cdot)$. Per la matrice di adiacenza.</i>	42
4.2	<i>Calcolo dell'errore come: $e_i^T f(\text{Lanczos})e_i - e_i^T f(A)e_i$. Come modulo della differenza tra il valore di Gauss e il valore: $\text{diag}(\text{expm}(A))$. Rete sprand(10) con 10 nodi.</i>	43
4.3	<i>Tempo di calcolo per rete sprand(1000), con 1000 nodi, applicando l'algoritmo iterativo Lanczos/Quadratura e confrontandolo con l'approssimazione di Padé $\text{expm}(A)$. Per la matrice di adiacenza.</i>	44
4.4	<i>Calcolo dell'errore come: $e_i^T f(\text{Lanczos})e_i - e_i^T f(A)e_i$. Come modulo della differenza tra il valore di Gauss e il valore: $\text{diag}(\text{expm}(A))$. Rete sprand(1000) con 1000 nodi.</i>	45
4.5	<i>Tempo di calcolo per rete sprand(2000), con 2000 nodi, applicando l'algoritmo iterativo Lanczos/Quadratura, confronto con l'approssimazione di Padé $\text{expm}(\cdot)$. Per la matrice di adiacenza.</i>	46

4.6	Calcolo dell'errore come: $ e_i^T f(\text{Lanczos})e_i - e_i^T f(A)e_i $. Come modulo della differenza tra il valore di Gauss e il valore: $\text{diag}(\text{expm}(A))$. Rete spran(2000) con 2000 nodi.	47
4.7	Tempo di calcolo per rete Yeast, con 2114 nodi e 4480 archi, applicando l'algoritmo iterativo Lanczos/Quadratura, confronto con l'approssimazione di Padé $\text{expm}(\cdot)$. Per la matrice di adiacenza.	48
4.8	Calcolo dell'errore come: $ e_i^T f(\text{Lanczos})e_i - e_i^T f(A)e_i $. Come modulo della differenza tra il valore di Gauss e il valore: $\text{diag}(\text{expm}(A))$. Rete Yeast con 2114 nodi e 4480 archi.	49
4.9	Tempo di calcolo per rete Power, con 4941 nodi e 13188 archi, applicando l'algoritmo iterativo Lanczos/Quadratura, confronto con l'approssimazione di Padé $\text{expm}(\cdot)$. Per la matrice di adiacenza.	50
4.10	Calcolo dell'errore come: $ e_i^T f(\text{Lanczos})e_i - e_i^T f(A)e_i $. Come modulo della differenza tra il valore di Gauss e il valore: $\text{diag}(\text{expm}(A))$. Rete Yeast con 4941 nodi e 13188 archi.	51

Capitolo 1

Preliminari

IN questo capitolo si concentrano dei risultati utili per la successiva trattazione nel testo. Nella prima parte si richiamano alcune definizioni, spazio lineare, rango etc. e le sue proprietà che le caratterizzano. Si analizzeranno le matrici che possiedono determinate strutture. Nella seconda parte si analizzerà la connessione tra gli Algoritmi di Lanczos e le formule di quadratura analizzate da Golub ed altri.

Spazi

Definizione 1.1.1: Spazio Lineare

Uno spazio lineare o vettoriale reale è un insieme V su cui sono definite due operazioni di somma e prodotto per uno scalare

$$\begin{aligned} + : V \times V &\longrightarrow V & \cdot : \mathbb{R} \times V &\longrightarrow V \\ (x, y) &\longmapsto x + y & (\alpha, x) &\longmapsto \alpha x \end{aligned}$$

che, per ogni $\alpha, \beta \in \mathbb{R}$ e $x, y, z \in V$ godono delle seguenti proprietà:

- 1 $x + y \in V$ (chiusura rispetto alla somma)
- 2 $\alpha x \in V$ (chiusura rispetto al prodotto)
- 3 $y + x = x + y$ (proprietà commutativa)
- 4 $(x + y) + z = x + (y + z)$ (proprietà associativa)
- 5 $\exists 0 \in V : x + 0 = x$ (elemento neutro)
- 6 $\exists -x \in V : x + (-x) = 0$ (inverso additivo)
- 7 $\alpha(\beta x) = (\alpha\beta)x$ (proprietà associativa)
- 8 $\alpha(x + y) = \alpha x + \alpha y$ (proprietà distributiva in V)

9 $(\alpha + \beta)x = \alpha x + \beta x$ (proprietà distributiva in \mathbb{R})

10 $1x = x$ (elemento neutro)

Si chiama **combinazione lineare** dei vettori $\{x_1, \dots, x_k\}$ con coefficienti $\alpha_i \in \mathbb{R}$, $i = 1, \dots, k$ il vettore $x \in V$ dato da

$$x = \sum_{i=1}^k \alpha_i x_i$$

sottospazio generato dai vettori $\{x_1, \dots, x_k\}$ è formato da tutte le loro combinazioni lineari

$$\text{span}(x_1, \dots, x_k) := \left\{ x \in V : x = \sum_{i=1}^k \alpha_i x_i \right\}.$$

I vettori $\{x_1, \dots, x_k\}$ sono **linearmente indipendenti** se

$$\sum_{i=1}^k \alpha_i x_i = 0 \Rightarrow \alpha_i = 0, i = 1, \dots, k.$$

Questa condizione equivale al fatto che nessuno dei vettori sia combinazione lineare degli altri.

Una **base** è un insieme di vettori linearmente indipendenti tali che ogni vettore dello spazio possa essere espresso come combinazione lineare.

La **dimensione** di uno spazio lineare è la cardinalità¹ di una base.

Definizione 1.1.2: Spazio normato

Uno **spazio normato** è uno spazio lineare su cui è definita una funzione

$$\begin{aligned} \|\cdot\| : V &\longrightarrow \mathbb{R} \\ x &\longmapsto \|x\| \end{aligned}$$

detta **norma**, tale che per ogni $x, y \in V$ e $\alpha \in \mathbb{R}$ si abbia

- 1 $\|x\| \geq 0$ e $\|x\| = 0 \iff x = 0$ (positività)
- 2 $\|\alpha x\| = |\alpha| \|x\|$ (omogeneità)
- 3 $\|x + y\| \leq \|x\| + \|y\|$ (diseguaglianza triangolare)

uno spazio normato è anche uno **spazio metrico**, in cui la distanza tra due vettori è misurata mediante la funzione:

$$d(x, y) := \|x - y\|.$$

¹ La cardinalità di un insieme finito è il numero dei suoi elementi. La definizione si può estendere anche agli insiemi infiniti, generalizzando il concetto di **numero naturale**.

Le più importanti norme in uno spazio \mathbb{R}^n sono:

$$\text{norma due: } \|x\|_2 = \left(\sum_{i=1}^n x_i^2 \right)^{\frac{1}{2}}$$

$$\text{norma uno: } \|x\|_1 = \sum_{i=1}^n |x_i|$$

$$\text{norma } \infty : \|x\|_\infty = \max_{i=1, \dots, n} |x_i|$$

Gli spazi funzionali $C[a, b]$ e $L^2[a, b]$ vengono dotati delle norme

$$\|f\|_\infty = \max_{x \in [a, b]} |f(x)|, \quad f \in C[a, b]$$

$$\|g\|_2 = \left(\int_a^b |g(x)|^2 dx \right)^{\frac{1}{2}}, \quad g \in L^2[a, b].$$

Un risultato importante che ci servirà di seguito è il seguente teorema:

Teorema 1.1.1: Equivalenza delle norme

Tutte le norme in \mathbb{R}^n sono equivalenti, nel senso che per ogni coppia di norme $\|\cdot\|_\alpha, \|\cdot\|_\beta$ esistono due costanti positive m e M tali che, per ogni $x \in \mathbb{R}^n$, si ha:

$$m\|x\|_\beta \leq \|x\|_\alpha \leq M\|x\|_\beta.$$

Questo teorema consente di studiare indifferentemente qualsiasi norma per la convergenza di una successione di vettori.

Definizione 1.1.3: Spazio di Hilbert

Uno spazio di Hilbert è uno spazio lineare su cui è definito un prodotto scalare (o prodotto interno)

$$\begin{aligned} \langle \cdot, \cdot \rangle : V \times V &\longrightarrow \mathbb{R} \\ (x, y) &\longmapsto \langle x, y \rangle \end{aligned}$$

tale che per ogni $x, y, z \in V$ e $\alpha \in \mathbb{R}$

- 1 $\langle x, x \rangle \geq 0$ e $\langle x, x \rangle = 0 \iff x = 0$ (positività)
- 2 $\langle x, y \rangle = \langle y, x \rangle$ (proprietà commutativa)
- 3 $\langle \alpha x, x \rangle = \alpha \langle x, x \rangle$ (omogeneità)
- 4 $\langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle$ (linearità)

Tale spazio deve essere completo rispetto alla norma indotta dal prodotto interno

$$\|x\| := \langle x, x \rangle^{\frac{1}{2}}$$

Due vettori sono *ortogonali* quando il loro prodotto scalare è nullo. Il prodotto scalare comunemente utilizzato in \mathbb{R}^n è: $\langle x, y \rangle = \sum_{i=1}^n x_i y_i$.

Matrici

Una matrice $m \times n$ è un quadro $m \times n$ di numeri reali o complessi di m righe e n colonne. Sia $\mathbf{A} \in \mathbb{C}^{m \times n}$ la matrice *aggiunta* \mathbf{A}^* si ottiene scambiando righe si \mathbf{A} con le sue colonne e coniugandone gli elementi, se $\mathbf{A} \in \mathbb{R}$ allora si dice *trasposta*. Date due matrici $\mathbf{A} \in \mathbb{R}^{m \times n}$ e $\mathbf{B} \in \mathbb{R}^{n \times p}$ il prodotto tra matrici è la matrice $\mathbf{C} = \mathbf{AB} \in \mathbb{R}^{m \times p}$ definita da:

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}, \quad i = 1, \dots, m; \quad j = 1, \dots, p.$$

Una matrice reale $m \times n$ rappresenta la generica *trasformazione lineare* tra gli spazi lineari \mathbb{R}^n e \mathbb{R}^m . Il Prodotto matriciale corrisponde alla composizione di due trasformazioni lineari.

Una matrice *quadrata* A si dice *invertibile* o *non singolare* se esiste una matrice \mathbf{A}^{-1} detta matrice inversa tale che $\mathbf{A}^{-1}\mathbf{A} = \mathbf{AA}^{-1} = \mathbf{I}$. Definiamo alcune proprietà delle matrici:

- $(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$;
- $(\mathbf{AB})^{-1} = \mathbf{B}^{-1} \mathbf{A}^{-1}$;
- $(\mathbf{A}^T)^{-1} = (\mathbf{A}^{-1})^T$;
- \mathbf{A} è invertibile se e solo se ha righe (colonne) linearmente indipendenti.

Il *determinante* è una funzione che associa a ciascuna matrice quadrata un numero reale, esso può essere calcolato mediante la formula di Laplace;

$$\det(\mathbf{A}) = \sum_{j=1}^n (-1)^{i+j} a_{ij} \det(A_{ij})$$

essendo A_{ij} la sotto-matrice che si ottiene sa A eliminando la i -esima riga e j -esima colonna. Alcune proprietà dei determinanti:

- $\det(\mathbf{A}^T) = \det(\mathbf{A}), \det(\mathbf{A}^*) = \overline{\det(\mathbf{A})}, \det(\mathbf{A}^{-1}) = \det(\mathbf{A})^{-1}$;
- $\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B}), \det(\alpha\mathbf{A}) = \alpha^n \det(\mathbf{A})$;
- uno scambio di due righe produce un cambio di segno del determinante;
- A è non singolare se e solo se $\det(\mathbf{A}) \neq 0$

Definizione 1.2.1: Rango

Il *rango* di una matrice $\text{rank}(\mathbf{A})$ può essere definito come il massimo numero di righe (o colonne) linearmente indipendenti o come l'ordine della più grande sotto-matrice quadrata con determinante non nullo (non singolare).

Autovalori e autovettori

Definizione 1.2.2: Autovalore e autovettore

Si dicono *autovalore* ed *autovettore* di una matrice \mathbf{A} uno scalare λ ed un vettore $\mathbf{x} \neq 0$ che verificano la relazione

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \quad (1.1)$$

La 1.1 può essere riscritta:

$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = 0.$$

Questo è un sistema lineare omogeneo dove il vettore $\mathbf{x} \neq 0$, pertanto per ammettere una soluzione non nulla deve annullarsi il determinante della matrice $(\mathbf{A} - \lambda\mathbf{I})$, questo risulta essere un polinomio di grado n in λ , dove n è il numero di righe della matrice \mathbf{A} quadrata. Tale polinomio è noto come *polinomio caratteristico*:

$$p_{\mathbf{A}}(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I})$$

quindi calcolare gli zeri di $p_{\mathbf{A}}(\lambda)$ vuol dire calcolare gli n autovalori di \mathbf{A} .

Per ciascun autovalore $\lambda_k, k = 1, \dots, n$, una soluzione non nulla del **sistema singolare** omogeneo²

$$(\mathbf{A} - \lambda_k\mathbf{I})\mathbf{x} = 0$$

fornisce il corrispondente autovettore, il rango della matrice $(\mathbf{A} - \lambda_k\mathbf{I}) \leq n - 1$, dunque l'autovettore rimane determinato a meno di una costante.

Definizione 1.2.3: Spettro

Lo spettro di una matrice è l'insieme degli autovalori

$$\sigma(\mathbf{A}) = \{\lambda_1, \dots, \lambda_n\} \quad (1.2)$$

Definizione 1.2.4: Raggio spettrale

Il raggio spettrale è il massimo dei moduli degli autovalori

$$\rho(\mathbf{A}) = \max_{k=1, \dots, n} |\lambda_k|. \quad (1.3)$$

La *molteplicità algebrica* di un autovalore è la sua molteplicità come zero del polinomio caratteristico. Si definisce, *molteplicità geometrica* di un autovalore il massimo numero di autovettori linearmente indipendenti ad esso corrispondenti, in generale per ogni autovalore si ha sempre:

$$\text{molteplicità geometrica} \leq \text{molteplicità algebrica}$$

Se per un autovalore vale il minore stretto, al matrice viene detta *difettiva*.

Proprietà:

²Il sistema è singolare $\det(\mathbf{A} - \lambda_k\mathbf{I}) = 0$, stiamo cercando soluzioni non banali del vettore \mathbf{x} .

- $\det(\mathbf{A}) = \prod_{k=1}^n \lambda_k$;
- $\sigma(\mathbf{A}^T) = \sigma(\mathbf{A}), \sigma(\mathbf{A}^{-1}) = \{\lambda_1^{-1}, \dots, \lambda_n^{-1}\}, \sigma(\mathbf{A}^p) = \{\lambda_1^p, \dots, \lambda_n^p\}$;
- **ad autovalori distinti corrispondono autovettori indipendenti;**
- se un autovettore \mathbf{x} è noto il **quoziente di Rayleigh**: $\frac{\mathbf{x}^T \mathbf{A} \mathbf{x}}{\mathbf{x}^T \mathbf{x}}$ fornisce il corrispondente autovalore.

Matrice Normale

Definizione 1.2.5: Matrice Normale

Una matrice quadrata $\mathbf{A} \in \mathbb{C}^{n \times n}$ è normale se $\mathbf{A}^* \mathbf{A} = \mathbf{A} \mathbf{A}^*$, dove $\mathbf{A}^* = \mathbf{A}^T$ è la coniugata trasposta di \mathbf{A} .

Decomposizione spettrale

Teorema 1.2.1: Decomposizione spettrale

Ponendo $\mathbf{A} \in \mathbb{C}^{n \times n}$ una matrice normale. Allora esiste una matrice unitaria $\mathbf{V} \in \mathbb{C}^{n \times n}$ e una matrice diagonale $\Lambda \in \mathbb{C}^{n \times n}$ tale che:

$$\mathbf{A} = \mathbf{V} \Lambda \mathbf{V}^* \quad (1.4)$$

dove Λ contiene gli autovalori di \mathbf{A} e le colonne di \mathbf{V} sono le corrispondenti (propriamente normalizzati) autovettori. La 1.4 è chiamata **Decomposizione Spettrale** della matrice \mathbf{A} .

Matrici di forma particolare

Alcune matrici posseggono una **struttura** quando hanno delle proprietà che rendono più agevole la risoluzione di un problema che le coinvolge, ecco alcune di esse.

Hermitiane

Una matrice è Hermitiana se coincide con la sua aggiunta, se la matrice è reale si dice simmetrica. Una matrice Hermitiana A è definita positiva **se**:

$$x^*Ax > 0, \quad \forall x \in \mathbb{C}^n$$

valida per matrici simmetriche reali

Teorema 1.3.1: Matrice Hermitiana

Se A è Hermitiana, i suoi autovalori sono reali ed esiste per \mathbb{C}^n una base di autovettori ortogonali. Se A è anche definita positiva, gli autovalori sono positivi.

Hessenberg

Una matrice di Hessenberg è una matrice del tipo:

$$\tilde{A} = \begin{bmatrix} * & \dots & \dots & * \\ * & \ddots & & \vdots \\ & \ddots & \ddots & \vdots \\ & & * & * \end{bmatrix} \quad (1.5)$$

dove $\tilde{a}_{ij} = 0$ per $i > j+1$. La forma di Hessenberg è invariante per un particolare tipo di trasformazioni (QR)³ nel senso che se la matrice A cui viene applicata la trasformazione è di Hessenberg, tali saranno tutte le matrici della successione da esso generata.

Unitarie

Una matrice complessa è unitaria se la sua inversa coincide con l'aggiunta:

$$QQ^* = Q^*Q = I$$

Una matrice reale è *ortogonale* se l'inverso coincide con la trasposta:

$$Q^T Q = Q Q^T = I$$

³Si veda [Rod08] per maggiori dettagli.

Teorema 1.3.2:

Se \mathbf{Q} è unitaria, allora:

- 1 $|\det(\mathbf{Q})| = 1$, se $\mathbf{Q} \in \mathbb{R}^{n \times n} \Rightarrow \det(\mathbf{Q}) = \pm 1$;
- 2 $\|\mathbf{Q}\mathbf{x}\|_2 = \|\mathbf{x}\|_2$, per ogni $\mathbf{x} \in \mathbb{C}^n$.

Triangolari

Una matrice \mathbf{U} si dice triangolare superiore se i suoi elementi verificano $u_{ij} = 0$, per $i > j$. Una matrice \mathbf{L} si dice triangolare inferiore se i suoi elementi verificano $l_{ij} = 0$, per $i < j$. Una matrice \mathbf{D} si dice diagonale se è simultaneamente triangolare superiore e inferiore.

Teorema 1.3.3:

Se \mathbf{T} è triangolare il suo determinante è dato dal prodotto degli elementi diagonali. Gli autovalori coincidono con gli elementi diagonali.

Le matrici triangolari formano un'algebra:

- 1 l'inversa di una matrice triangolare, quando esiste;
- 2 il prodotto di due matrici triangolari;

sono ancora matrici triangolari dello stesso tipo.

La stessa proprietà vale per matrici unitarie, ma questo caso non si può parlare di algebra dato che l'unitarietà non si trasmette attraverso la somma di matrici ed il prodotto per uno scalare.

Banda

Gli elementi di una matrice a banda verificano la relazione:

$$a_{ij} = 0, \text{ se } i - j \geq k \text{ oppure } i - j \leq -m$$

Sparse

Le matrici sparse sono dotate di pochi elementi diversi da zero, meno del 10%. Pertanto si possono memorizzare solo gli elementi non nulli e della loro posizione all'interno della matrice.

Norme matriciali

Lo spazio lineare $\mathcal{M}_{m \times n}$ cui appartengono le matrici, può essere dotato della struttura di spazio normato definendo una qualsiasi norma $\|\cdot\|$ che verifichi gli assiomi richiesti.

Definizione 1.3.1: Submoltiplicativa

Una norma matriciale è **submoltiplicativa** se per ogni coppia di matrici di dimensioni compatibili si ha

$$\|\mathbf{AB}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{B}\|.$$

Definizione 1.3.2: Consistente

Una norma di matrice è **consistente** con le norme vettoriali $\|\cdot\|_a$ di \mathbb{R}^n e $\|\cdot\|_b$ di \mathbb{R}^m se

$$\|\mathbf{Ax}\|_b \leq \|\mathbf{A}\| \cdot \|\mathbf{x}\|_a, \forall \mathbf{A} \in \mathbb{R}^{m \times n}, \forall \mathbf{x} \in \mathbb{R}^n.$$

Spesso le norme $\|\cdot\|_a$ e $\|\cdot\|_b$ coincidono, in tal caso il pedice si trascura.

Definizione 1.3.3: Norma indotta

Una norma matriciale si dice **indotta** da una norma vettoriale $\|\cdot\|$, se:

$$\|\mathbf{A}\| = \sup_{\mathbf{x} \neq 0} \frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|}$$

o se, equivalentemente,

$$\|\mathbf{A}\| = \max_{\|\mathbf{x}\|=1} \|\mathbf{Ax}\|.$$

Una norma così definita viene anche detta **naturale**, o subordinata.

Seguono alcuni teoremi sulle norme matriciali

Teorema 1.3.4: Norma matriciale indotta dalla norma vettoriale ∞

$$\|\mathbf{A}\|_\infty = \max_{i=1, \dots, m} \sum_{j=1}^n |a_{ij}|.$$

Teorema 1.3.5: Norma matriciale indotta dalla norma 1

$$\|\mathbf{A}\|_1 = \max_{i=1, \dots, n} \sum_{j=1}^m |a_{ij}|.$$

Teorema 1.3.6: Norma matriciale indotta dalla norma 2

$$\|\mathbf{A}\|_2 = \sqrt{\rho(\mathbf{A}^* \mathbf{A})}.$$

Né consegue che se \mathbf{A} è Hermitiana:

$$\|\mathbf{A}\|_2 = \rho(\mathbf{A}).$$

Forma canonica di Jordan

Definizione 1.3.4: Forma canonica di Jordan

La **forma di Jordan** di una matrice quadrata $\mathbf{A} \in \mathbb{C}^{n \times n}$ è una matrice $\mathbf{J} \in \mathbb{C}^{n \times n}$ tale che

$$\mathbf{Z}^{-1} \mathbf{A} \mathbf{Z} = \mathbf{J} = \text{diag}(\mathbf{J}_1, \dots, \mathbf{J}_p) \quad (1.6)$$

dove

$$\mathbf{J}_k = \begin{bmatrix} \lambda_k & 1 & & \\ & \lambda_k & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_k \end{bmatrix} \in \mathbb{C}^{m_k \times m_k}$$

Numero di condizionamento

Un parametro di fondamentale importanza, per la soluzione di un sistema lineare è il *numero di condizionamento*.

Definizione 1.3.5: Numero di condizionamento

Il numero di condizionamento di una matrice relativamente alla risoluzione di un sistema lineare, è la quantità:

$$k(\mathbf{A}) = \|\mathbf{A}\| \|\mathbf{A}^{-1}\|. \quad (1.7)$$

Misura il massimo fattore di amplificazione dell'errore relativo sulla soluzione rispetto all'errore relativo sui dati. Esso è influenzato dalla norma matriciale adottata.

Verifica le seguenti proprietà:

- ❶ per ogni matrice \mathbf{A} , $k(\mathbf{A}) \geq 1$;
- ❷ per ogni norma naturale, $k(\mathbf{I}) = 1$;
- ❸ $k(\mathbf{AB}) \leq k(\mathbf{A})k(\mathbf{B})$;

4 se \mathbf{Q} è ortogonale $k_2(\mathbf{Q}) = 1$;

5 $k_2(\mathbf{A}) = \sqrt{\frac{\lambda_{\max}(\mathbf{A}^T \mathbf{A})}{\lambda_{\min}(\mathbf{A}^T \mathbf{A})}}$;

6 se A è simmetrica $k_1(\mathbf{A}) = \frac{|\lambda_{\max}(\mathbf{A})|}{|\lambda_{\min}(\mathbf{A})|}$.

Ortogonalizzazione di Gram-Schmidt

Il processo di Gram-Schmidt consente di ortogonalizzare n fattori di \mathbb{R}^m con ($m \geq n$). Iniziamo pensando di poter scrivere la matrice \mathbf{A} come composta da vettori colonna:

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$$

allora questo procedimento permette di scrivere:

$$\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n]$$

tale che

$$\text{span}\{\mathbf{q}_1, \dots, \mathbf{q}_n\} = \text{span}\{\mathbf{a}_1, \dots, \mathbf{a}_n\} \quad \text{e} \quad \mathbf{q}_i^T \mathbf{q}_j = \delta_{ij}.$$

Dove:

$$\delta_{ij} := \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j. \end{cases} \quad (1.8)$$

Al primo passo si pone:

$$\tilde{\mathbf{q}}_1 = \mathbf{a}_1, \quad r_{11} = \|\tilde{\mathbf{q}}_1\|_2 \quad \mathbf{q}_1 = \frac{\tilde{\mathbf{q}}_1}{r_{11}}.$$

Al generico passo k si calcola

$$\tilde{\mathbf{q}}_k = \mathbf{a}_k - \sum_{j=1}^{k-1} r_{jk} \mathbf{q}_j \quad (1.9)$$

con

$$r_{jk} = \mathbf{q}_j^T \mathbf{a}_k \quad j = 1, 2, \dots, k-1,$$

ottenendo

$$\mathbf{q}_k = \frac{\tilde{\mathbf{q}}_k}{r_{kk}}, \quad \text{per } r_{kk} = \|\tilde{\mathbf{q}}_k\|.$$

La scelta effettuata per le costanti r_{jk} permette che ogni vettore \mathbf{q}_k risulti ortogonale ai precedenti.

Applicando l'algoritmo, pedissequamente, per esempio al passo $k = 3$ si ha.

$$k = 3 \quad \tilde{\mathbf{q}}_3 = \mathbf{a}_3 - r_{13} \mathbf{q}_1 - r_{12} \mathbf{q}_2 \quad (1.10)$$

$$r_{13} = \mathbf{q}_1^T \mathbf{a}_3$$

$$r_{12} = \mathbf{q}_1^T \mathbf{a}_2$$

$$\mathbf{q}_3 = \frac{\tilde{\mathbf{q}}_3}{r_{33}} \quad (1.11)$$

e unendo le 1.10 e 1.11 si ottiene:

$$\mathbf{q}_3 = \frac{\mathbf{a}_3 - r_{13} \mathbf{q}_1 - r_{12} \mathbf{q}_2}{r_{33}} \Rightarrow \mathbf{a}_3 = r_{13} \mathbf{q}_1 + r_{12} \mathbf{q}_2 + r_{33} \mathbf{q}_3,$$

che possiamo riscrivere come:

$$\mathbf{a}_k = \sum_{j=1}^k r_{jk} \mathbf{q}_j, \quad k = 1, \dots, n \quad (1.12)$$

e dunque la matrice \mathbf{A} può essere rappresentata come:

$$\mathbf{A} = [\mathbf{q}_1 \dots \mathbf{q}_k \dots \mathbf{q}_n] \begin{bmatrix} r_{11} & \dots & r_{1k} & \dots & r_{1n} \\ & \ddots & \vdots & & \vdots \\ & & r_{kk} & & \vdots \\ & & & \ddots & \vdots \\ & & & & r_{nn} \end{bmatrix}.$$

L'ortogonalizzazione delle colonne della matrice \mathbf{A} costruisce la fattorizzazione QR. Questo procedimento è noto come **CGS** (*Classical Gram-Schmidt*). La matrice \mathbf{R} è costruita per colonne e quando $n < m$ la matrice \mathbf{Q} pur avendo colonne ortonormali, **non** è ortogonale, non essendo quadrata. Può essere completata per ottenere una matrice ortogonale aggiungendo $m - n$ vettori che completino la base ortonormale.

È stata studiata una differente versione dell'algoritmo, perché la colonne di \mathbf{Q} tendono a perdere la reciproca ortogonalità al procedere delle iterazioni, rendendo numericamente instabile l'algoritmo.

La versione prende il nome di **MGS** (*Modified Gram-Schmidt*), genera una successione di $n + 1$ matrici tali che $\mathbf{A}^{(1)} = \mathbf{A}$ e $\mathbf{A}^{(n+1)} = \mathbf{Q} \in \mathbb{R}^{m \times n}$.

La k -esima matrice assume la forma

$$\mathbf{A}^{(k)} = [\mathbf{q}_1, \dots, \mathbf{q}_{k-1}, \mathbf{a}_k^{(k)}, \dots, \mathbf{a}_n^{(k)}]$$

con

$$\begin{cases} \mathbf{q}_i^T \mathbf{q}_j = \delta_{ij}, & i, j = 1, \dots, k-1, \\ \mathbf{q}_j^T \mathbf{a}_j^{(k)} = 0, & i = 1, \dots, k-1; j = k, \dots, n. \end{cases}$$

Al passo k , la matrice $\mathbf{A}^{(k+1)}$ viene ottenuta attraverso la seguente trasformazione:

$$[\mathbf{q}_1, \dots, \mathbf{q}_{k-1}, \mathbf{a}_k^{(k)}, \dots, \mathbf{a}_n^{(k)}] \rightarrow [\mathbf{q}_1, \dots, \mathbf{q}_k, \mathbf{a}_{k+1}^{(k+1)}, \dots, \mathbf{a}_n^{(k+1)}]$$

ponendo

$$\begin{aligned} r_{kk} &= \|\mathbf{a}_k^{(k)}\|, \\ \mathbf{q}_k &= \frac{\mathbf{a}_k^{(k)}}{r_{kk}}, \\ r_{kj} &= \mathbf{q}_k^T \mathbf{a}_j^{(k)}, \quad j = k+1, \dots, n, \\ \mathbf{a}_j^{(k+1)} &= \mathbf{a}_j^{(k)} - r_{jk} \mathbf{q}_k, \quad j = k+1, \dots, n. \end{aligned}$$

Si ottengono due famiglie di vettori:

- ① $\{q_1, \dots, q_k\}$ formata da vettori tra loro ortonormali,
- ② $\{a_{k+1}^{(k+1)}, \dots, a_n^{(k+1)}\}$ gli elementi sono ortogonali a tutti i vettori della prima famiglia.

All' n -esima iterazione la prima famiglia conterrà n vettori tra loro ortonormali, la seconda sarà vuota. L'algoritmo termina a n passi, la matrice \mathbf{R} viene costruita per righe, anziché per colonne. La complessità computazionale è $O(mn^2)$. Anche in **MGS** può verificarsi una perdita di ortogonalità nelle colonne di \mathbf{A} .

Polinomi ortogonali

Definizione 1.5.1: Integrale di Riemann-Stieltjes

L'integrale di Riemann-Stieltjes (o semplicemente l'integrale di Stieltjes) di una funzione $f: \mathbb{R} \rightarrow \mathbb{R}$ rispetto ad una funzione $\omega: \mathbb{R} \rightarrow \mathbb{R}$ è definito come:

$$\int_a^b f(x) d\omega(x) = \lim_{\delta(P) \rightarrow 0} \sum_{i=0}^{n-1} f(c_i) (\omega(x_{i+1}) - \omega(x_i)), \quad c_i \in [x_i, x_{i+1}]$$

dove $P = \{a = x_0 < x_1 < \dots < x_n = b\}$ è una partizione dell'intervallo $[a, b]$ e

$$\delta(P) = \max_{x_i \in P} |x_{i+1} - x_i|.$$

Le due funzioni f e ω sono chiamate integrando e integratore rispettivamente.

Nel seguito useremo anche la notazione

$$\int_a^b f(x) w(x) dx,$$

dove $w(x)$ è la funzione peso, una funzione differenziabile.

Definizione 1.5.2: Prodotto interno

Sia Π_n lo spazio di polinomi reali di grado non superiore a n . Dati due polinomi $p, q \in \Pi_n$ il prodotto interno rispetto alla misura ω è definito come:

$$\langle p, q \rangle = \int_a^b p(x) q(x) d\omega(x), \quad (1.13)$$

e la norma di p è definita come:

$$\|p\|_\omega = \left(\int_a^b p(x)^2 d\omega(x) \right)^{\frac{1}{2}}.$$

La versione discreta del prodotto interno è data da:

$$\langle p, q \rangle = \sum_{j=1}^m p(x_j)q(x_j)w_j^2. \quad (1.14)$$

I valori x_j e w_j sono chiamati nodi e pesi, rispettivamente. La relazione 1.14 può essere vista come approssimazione della 1.13, oppure possiamo vedere l'integrale di Stieltjes rispetto alla misura $\omega(x)$ definito come:

$$\omega(x) = \begin{cases} 0 & \text{se } x < t_1, \\ \sum_{j=1}^i w_j^2 & \text{se } t_i \leq x < t_{i+1}, i = 1, \dots, m-1, \\ \sum_{j=1}^m w_j^2 & \text{se } x \geq t_m. \end{cases} \quad (1.15)$$

Lo spazio $(L_{\omega}^2, \langle \cdot, \cdot \rangle)$ di funzioni misurabili tale che:

$$\int_a^b f(x)^2 d\omega(x) < \infty$$

con prodotti interi definiti nelle 1.13 e 1.14 è uno spazio di Hilbert, si veda la definizione di spazio di Hilbert a pagina 3.

Definizione 1.5.3: Polinomi ortogonali

Un insieme di polinomi $\{p_i\}, i = 1, 2, \dots$, è detto **ortogonale** se $\langle p_i, p_j \rangle = 0 \forall i \neq j$. Inoltre i polinomi sono ortonormali se $\langle p_i, p_j \rangle = \delta_{ij}$, si veda la 1.8 a pagina 12.

Teorema 1.5.1: Tre termini di ricorrenza

Per ogni famiglia di polinomi ortogonali esiste una sequenza di coefficienti α_k e $\beta_k, k = 1, 2, \dots$, tali che

$$\begin{aligned} \beta_{k+1}p_{k+1}(x) &= (x - \alpha_{k+1})p_k(x) - \beta_k p_{k-1}(x), \quad k = 0, 1, \dots \quad (1.16) \\ p_1 &:= 0, \quad \text{e} \quad p_0 := \beta_0 \end{aligned}$$

con $\alpha_{k+1} = \langle xp_k, p_k \rangle, k = 0, 1, \dots$, e β_k tale che $\|p_k\|_{\omega} = 1$.

Se il polinomio ortonormale esiste per ogni k , allora possiamo associare ad esso una matrice tridiagonale simmetrica, chiamata la **matrice di Jacobi** definita come:

$$\mathbf{J}_{\infty} = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \beta_3 & \\ & & \ddots & \ddots & \ddots \end{bmatrix} \quad (1.17)$$

Chiamando \mathbf{J}_k la sottomatrice principale di ordine k . La relazione di ricorrenza 1.16 può essere scritta in forma matriciale come

$$xP_k(x) = \mathbf{J}_k P_k(x) + \beta_k p_k(x) \mathbf{e}_k \quad (1.18)$$

dove $P_k(x) = [p_0(x), p_1(x), \dots, p_{k-1}(x)]^T$ è il vettore del primo k polinomio ortonormale valutato in x , \mathbf{J}_k è la matrice di Jacobi di ordine k e \mathbf{e}_k è l'ultima colonna della matrice identità di ordine k .

Teorema 1.5.2:

Gli zeri $\theta_j^{(k)}$ del polinomio ortonormale p_k sono gli autovalori della matrice \mathbf{J}_k , chiamati **valori di Ritz** e $P_k(\theta_j^{(k)})$ sono i corrispondenti (non normalizzati) autovettori.

Dimostrazione. Per sostituzione diretta nella 1.18:

$$\begin{aligned} \theta_j^{(k)} P_k(\theta_j^{(k)}) &= \mathbf{J}_k P_k(\theta_j^{(k)}) \\ (\mathbf{J}_k - \theta_j^{(k)} \mathbf{I}) P_k(\theta_j^{(k)}) &= 0 \end{aligned}$$

□

Definizione 1.5.4: U

n polinomio matriciale $p(x)$ di ordine k , $x \in \mathbb{R}$ è definito come

$$p(x) = \sum_{j=0}^i x^j C_j$$

dove i coefficienti C_j sono matrici quadrate di ordine k

Definizione 1.5.5: Prodotto interno

Il prodotto interno di due polinomi matriciali p e q è definito come

$$\langle p, q \rangle = \int_a^b p(x) d\omega(x) q(x)^T \quad (1.19)$$

dove le misure $\omega(x)$ sono matrici quadrate di ordine k .

Definizione 1.5.6: Polinomi matriciali ortogonali

Una sequenza di polinomi matriciali $\{p_l\}, l = 0, 1, 2, \dots$ è ortonormale se

$$\langle p_i, p_j \rangle = \delta_{i,j} \mathbf{I}_k.$$

per ogni coppia (i, j) .

Iterazioni in sottospazi di Krylov

Definizione 1.6.1: P

rendendo una matrice simmetrica \mathbf{A} di ordine n e \mathbf{u} un vettore dato di ordine n . Un **sottospazio di Krylov** di ordine k è un sottospazio lineare delle immagini di \mathbf{u} e delle prime $k - 1$ potenze di \mathbf{A} tal che:

$$\mathcal{K}_r := \text{span}(\mathbf{u}, \mathbf{A}\mathbf{u}, \dots, \mathbf{A}^{r-1}\mathbf{u}) \subseteq \mathbb{R}^n.$$

La matrice

$$\mathbf{K}_k = [\mathbf{u}, \mathbf{A}\mathbf{u}, \mathbf{A}^2\mathbf{u}, \dots, \mathbf{A}^{k-1}\mathbf{u}]$$

è chiamata matrice di Krylov.

All'aumentare di k le colonne di \mathbf{K}_k tendono ad allinearsi con l'autovettore dominante di \mathbf{A} , rendendo \mathbf{K}_k mal condizionata.

I **metodi di proiezione in sottospazi di Krylov** determinano, per ogni k , una soluzione approssimata $\mathbf{x}^{(k)}$ del sistema lineare $\mathbf{A}\mathbf{x} = \mathbf{u}$ che appartenga al sottospazio \mathcal{K}_k . La scelta del vettore $\mathbf{x}^{(k)}$ può essere fatta applicando diversi criteri di ottimalità, che conducono a differenti algoritmi. Se i vettori $\mathbf{A}^k\mathbf{u}, k = 0, 1, \dots, n - 1$, risultassero tutti indipendenti si avrebbe

$$\mathcal{K}_n \equiv \mathbb{R}^n$$

e l'algoritmo terminerebbe in n passi. Se questa condizione non fosse verificata potrebbe avvenire che per un dato $l < n$ il vettore $\mathbf{A}^l\mathbf{u}$ risulti linearmente dipendente dai vettori precedenti. In questo caso esisterebbero scalari $\alpha_i, i = 0, \dots, l$ non tutti nulli tali che

$$\sum_{i=0}^l \alpha_i \mathbf{A}^i \mathbf{u} = \mathbf{0}.$$

Supponendo $\alpha_0 \neq 0$, è possibile esplicitare in tale relazione il vettore \mathbf{u} , ottenendo:

$$\mathbf{A}\mathbf{x}^* = \mathbf{u} = -\frac{1}{\alpha_0} \sum_{i=1}^l \alpha_i \mathbf{A}^i \mathbf{u} = \mathbf{A} \left(-\frac{1}{\alpha_0} \sum_{i=1}^l \alpha_i \mathbf{A}^{i-1} \mathbf{u} \right),$$

dove \mathbf{x}^* indica la soluzione esatta del sistema lineare,

$$\mathbf{x}^* = -\frac{1}{\alpha_0} \sum_{i=1}^l \alpha_i \mathbf{A}^{i-1} \mathbf{b} \in \mathcal{K}_l$$

il che implica che il metodo terminerebbe in l passi. Questi metodi terminando a n passi vengono comunque considerati iterativi (*pseudo-iterativi*) in quanto, se opportunamente preconditionati forniscono una soluzione sufficientemente precisa in un numero di passi inferiore ad n .

L'iterazione di Arnoldi

L'iterazione di Arnoldi consente di costruire una base $\{\mathbf{q}_1, \dots, \mathbf{q}_l\}$ per il sottospazio \mathcal{K}_l formata da vettori tra loro ortonormali, cioè tali che $\mathbf{q}_i^T \mathbf{q}_j = \delta_{ij}$, sostituendo così la base numericamente instabile $\{\mathbf{b}, \mathbf{A}\mathbf{b}, \dots, \mathbf{A}^{l-1}\mathbf{b}\}$. Ciò viene fatto applicando il metodo di ortogonalizzazione di Gram-Schmidt, sezione 1.4:

$$\begin{aligned} \mathbf{q}_1 &= \frac{\mathbf{b}}{\|\mathbf{b}\|}, \\ \tilde{\mathbf{q}}_{s+1} &= \mathbf{A}^s \mathbf{b} - \sum_{i=1}^s h_{is} \mathbf{q}_i, \quad s = 1, \dots, l-1 \\ \mathbf{q}_{s+1} &= \frac{\tilde{\mathbf{q}}_{s+1}}{\|\tilde{\mathbf{q}}_{s+1}\|} \end{aligned}$$

dove le norme vettoriali utilizzate sono euclidee e gli scalari h_{is} possono essere ricavati imponendo l'ortogonalità di $\tilde{\mathbf{q}}_{s+1}$ con i vettori precedentemente calcolati. Per motivi numerici, per la costruzione di $\tilde{\mathbf{q}}_{s+1}$ si utilizza il vettore $\mathbf{A}\mathbf{q}_s$ in luogo di $\mathbf{A}^s \mathbf{b}$. Infatti, pur trattandosi di due differenti vettori, questa scelta consente di passare da una base di \mathcal{K}_s ad una di \mathbf{K}_{s+1} mediante un algoritmo più stabile. Si giunge così alle seguenti formule, che consentono di costruire una base ortonormale per \mathbf{K}_{l+1} :

$$\begin{aligned} \mathbf{q}_1 &= \frac{\mathbf{b}}{\|\mathbf{b}\|}, \\ \tilde{\mathbf{q}}_{s+1} &= \mathbf{A}\mathbf{q}_s - \sum_{i=1}^s h_{is} \mathbf{q}_i, \quad s = 1, \dots, l \\ \mathbf{q}_{s+1} &= \frac{\tilde{\mathbf{q}}_{s+1}}{h_{s+1,s}}, \quad s = 1, \dots, l, \\ h_{is} &= \mathbf{q}_i^T \mathbf{A}\mathbf{q}_s, \quad i = 1, \dots, s, \\ h_{s+1,s} &= \|\tilde{\mathbf{q}}_{s+1}\|. \end{aligned}$$

Di seguito l'Algoritmo:

Algoritmo: Iterazione di Arnoldi

Input: $\mathbf{A}, \mathbf{b}, \tau$

Output:

- 1 $\mathbf{q}_1 = \mathbf{b}/\|\mathbf{b}\|$
- 2 $s = 1$
- 3 repeat
 - $\mathbf{v} = \mathbf{A}\mathbf{q}_s$
 - for $i = 1, \dots, s$
 - ♣ $h_{is} = \mathbf{q}_i^T \mathbf{v}$
 - ♣ $\mathbf{v} = \mathbf{v} - h_{is}\mathbf{q}_i$
 - $h_{s+1,s} = \|\mathbf{v}\|$
 - $\mathbf{q}_{s+1} = \mathbf{v}/h_{s+1,s}$
 - $s = s + 1$
- 4 until $s = n + 1$ OR $|h_{s,s-1}| < \tau$

L'annullarsi, per un dato $l < n$, della norma $h_{l+1,l}$ interrompe l'algoritmo si parla allora di *breakdown*, e indica che il vettore $\mathbf{A}\mathbf{q}_l$ è **linearmente dipendente** dai precedenti e pertanto lo spazio \mathcal{K}_l contiene la soluzione del sistema lineare $\mathbf{A}\mathbf{x} = \mathbf{b}$.

Una delle formule trovate può essere riscritta come:

$$\mathbf{A}\mathbf{q}_s = \sum_{i=1}^{s+1} h_{i,s}\mathbf{q}_i, \quad s = 1, \dots, l,$$

esplicitando:

$$\begin{aligned} s = 1 \quad \mathbf{A}\mathbf{q}_1 &= \sum_{i=1}^2 h_{i,1}\mathbf{q}_i = h_{1,1}\mathbf{q}_1 + h_{2,1}\mathbf{q}_2 \\ s = 2 \quad \mathbf{A}\mathbf{q}_2 &= \sum_{i=1}^3 h_{i,2}\mathbf{q}_i = h_{1,2}\mathbf{q}_1 + h_{2,2}\mathbf{q}_2 + h_{3,2}\mathbf{q}_3 \\ &\vdots \\ s = l \quad \mathbf{A}\mathbf{q}_l &= \sum_{i=1}^{l+1} h_{i,l}\mathbf{q}_i = h_{1,l}\mathbf{q}_1 + h_{2,l}\mathbf{q}_2 + \dots + h_{l+1,l}\mathbf{q}_{l+1} \end{aligned}$$

si vede che può essere riscritta in maniera compatta:

$$\mathbf{A}\mathbf{Q}_l = \mathbf{Q}_{l+1}\tilde{\mathbf{H}}_l. \quad (1.20)$$

Dove $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_l] \in \mathbb{R}^{n \times l}$ e

$$\tilde{\mathbf{H}}_l = \begin{bmatrix} h_{1,1} & h_{1,2} & \dots & \dots & h_{1,l} \\ h_{2,1} & h_{2,2} & \dots & \dots & h_{2,l} \\ & h_{3,2} & \ddots & & h_{3,l} \\ & & \ddots & \ddots & \vdots \\ & & & h_{l,l-1} & h_{l,l} \\ & & & & h_{l+1,l} \end{bmatrix} \in \mathbb{R}^{(l+1) \times l}.$$

Dalla 1.20 si ottiene:

$$\mathbf{Q}_l^T \mathbf{A} \mathbf{Q}_l = \mathbf{H}_l := \begin{bmatrix} h_{1,1} & h_{1,2} & \dots & \dots & h_{1,l} \\ h_{2,1} & h_{2,2} & \dots & \dots & h_{2,l} \\ & h_{3,2} & \ddots & & h_{3,l} \\ & & \ddots & \ddots & \vdots \\ & & & h_{l,l-1} & h_{l,l} \end{bmatrix} \in \mathbb{R}^{(l) \times l}.$$

Che per $l = n$ diventa:

$$\mathbf{Q}_n^T \mathbf{A} \mathbf{Q}_n = \mathbf{H}_n,$$

l'iterazione di Arnoldi porta alla matrice di Hessenberg mediante n trasformazioni ortogonali. Essendo le matrici \mathbf{A} e \mathbf{H}_n unitariamente simili, questo potrebbe essere visto come un passo preliminare all'applicazione di un metodo per la ricerca degli autovalori che tragga vantaggio dalla struttura di Hessenberg.

In realtà la matrice \mathbf{H}_l contiene importanti informazioni sulla matrice \mathbf{A} anche per l molto minore di n . In molti casi gli autovalori estremali di \mathbf{H}_l convergono rapidamente agli autovalori estremali di \mathbf{A} e continuando con le iterazioni un numero sempre maggiore di autovalori delle due matrici tende a coincidere. Questo consente, nel caso di matrici di grandi dimensioni di ottenere dopo *poche* iterazioni approssimazioni sufficientemente accurate degli autovalori estremali dalla matrice in esame, i quali spesso consentono di determinare importanti informazioni sulla matrice stessa (raggio spettrale, condizionamento, etc.) o sul problema fisico da cui la matrice deriva.

L'iterazione di Lanczos

Supponiamo di applicare l'iterazione di Arnoldi ad una matrice \mathbf{A} simmetrica. Per questo particolare caso si ricava, che la matrice $\mathbf{H}_l = \mathbf{Q}_l^T \mathbf{A} \mathbf{Q}_l$, detta matrice di *Ritz*, dovendo essere simultaneamente simmetrica e in forma di Hessenberg, deve necessariamente essere tridiagonale. Essa sarà tale quindi che i suoi autovalori, i *valori di Ritz*, sono reali. Il metodo di Lanczos consente di ottenere una fattorizzazione $\mathbf{A} \mathbf{Q}_l = \mathbf{Q}_{l+1} \tilde{\mathbf{H}}_l$ dove $\mathbf{Q}_l = [\mathbf{q}_1, \dots, \mathbf{q}_l] \in \mathbb{R}^{n \times l}$ è costituita da colonne ortonormali che costituiscono una base per il sottospazio di Krylov di dimensione l .

Ponendo $h_{i,i} = \alpha_i$ e $h_{i+1,i} = h_{i,i+1} = \beta_i$ possiamo scrivere la matrice \mathbf{H}_l :

$$\mathbf{H}_l = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & & \beta_{l-1} & \\ & & & & \alpha_l \end{bmatrix} \quad (1.21)$$

Lo sfruttamento della particolare struttura della matrice porta ad un abbattimento della complessità computazionale. Il metodo, che prende il nome di **iterazione di Lanczos**, assume la forma:

$$\begin{aligned} \mathbf{q}_1 &= \frac{\mathbf{b}}{\|\mathbf{b}\|}, \\ \tilde{\mathbf{q}}_{s+1} &= \mathbf{A}\mathbf{q}_s - \alpha_s\mathbf{q}_s - \beta_{s-1}\mathbf{q}_{s-1}, \quad s = 1, \dots, l \\ \mathbf{q}_{s+1} &= \frac{\tilde{\mathbf{q}}_{s+1}}{\beta_s}, \quad s = 1, \dots, l, \\ \alpha_s &= \mathbf{q}_s^T \mathbf{A} \mathbf{q}_s, \\ \beta_s &= \|\tilde{\mathbf{q}}_{s+1}\|. \end{aligned}$$

Nell'algorithmo, se si dovesse verificare, per un certo passo $s < n$ un *breakdown* in output avrà le matrici \mathbf{Q} e \mathbf{H} ridotte e cioè di dimensioni $n \times s$ e $s \times s$ rispettivamente, dove s è l'indice dove si verifica il breakdown. Il presentarsi del breakdown nel caso dell'applicazione del metodo alla risoluzione di un sistema lineare $\mathbf{A}\mathbf{x} = \mathbf{b}$ implica che il sottospazio di Krylov \mathcal{K}_l conterrà la soluzione esatta del sistema lineare simmetrico.

Algoritmo: Iterazione di Lanczos

Input: $\mathbf{A}, \mathbf{b}, \tau$

Output:

- 1 $\beta_0 = 0, \mathbf{q}_0 = 0$
- 2 $\mathbf{q}_1 = \mathbf{b}/\|\mathbf{b}\|$
- 3 $s = 1$
- 4 repeat
 - $\mathbf{v} = \mathbf{A}\mathbf{q}_s$
 - $\alpha_s = \mathbf{q}_s^T \mathbf{v}$
 - $\mathbf{v} = \mathbf{v} - \alpha_s\mathbf{q}_s - \beta_{s-1}\mathbf{q}_{s-1}$
 - $\beta_s = \|\mathbf{v}\|$
 - $\mathbf{q}_{s+1} = \mathbf{v}/\beta_s$
 - $s = s + 1$
- 5 until $s = n + 1$ OR $|h_{s,s-1}| < \tau$

Dopo n steps l'algoritmo di Arnoldi termina nella fattorizzazione:

$$\mathbf{A} = \mathbf{U}_n \mathbf{T}_n \mathbf{U}_n^T$$

ovvero la matrice \mathbf{A} è simile alla matrice tridiagonale.

Teorema 1.6.1:

Il vettore di Lanczos \mathbf{u}_k può essere rappresentato come polinomio in \mathbf{A} applicando il vettore iniziale $\mathbf{u}_1 = \mathbf{u}$. In particolare il vettore $\mathbf{u}_{k+1} = p_k(\mathbf{A})\mathbf{u}_1$ dove:

$$p_k(\lambda) = (-1)^{k-1} \frac{\det(\mathbf{T}_k - \lambda \mathbf{I})}{\beta_1 \dots \beta_{k-1}}, \quad k \geq 1 \quad \beta_0 p_0(\lambda) \equiv 1 \quad (1.22)$$

La 1.22 soddisfa la ricorrenza a tre termini

$$\beta_{k+1} p_{k+1}(\lambda) = (\lambda - \alpha_{k+1}) p_k(\lambda) - \beta_k p_{k-1}(\lambda)$$

dove: $p_{-1} := 0$ e $p_0 := 1/\beta_0$.

Funzioni matriciali o di matrice

In algebra lineare si può estendere il concetto di funzione (matematica) alle matrici quadrate di qualsiasi ordine n attraverso l'associazione di una serie di MacLaurin ad ogni funzione, riducendola a una somma infinita di potenze di matrici:

$$f(A) = \sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!} A^k$$

Da cui risulta che una funzione di matrice quadrata è una matrice dello stesso ordine i cui elementi sono costituiti da una combinazione lineare della funzione degli elementi della matrice di partenza.

Ci sono molti modi equivalenti per definire una funzione di matrice, uno di questo è quello che usa la forma canonica di Jordan, si veda 1.6 a pagina 10.

Definizione 1.7.1: Funzione di matrice

Prendendo f definita nello spettro di $\mathbf{A} \in \mathbb{C}^{n \times n}$ e data \mathbf{A} una matrice in forma canonica di Jordan 1.6. Allora

$$f(\mathbf{A}) := \mathbf{Z} f(\mathbf{J}) \mathbf{Z}^{-1} = \mathbf{Z} \text{diag}(f(\mathbf{J}_k)) \mathbf{Z}^{-1} \quad (1.23)$$

dove

$$f(\mathbf{J}_k) = \begin{bmatrix} f(\lambda_k) & f'(\lambda_k) & \dots & \frac{f^{(m_k-1)}(\lambda_k)}{(m_k-1)!} \\ & f(\lambda_k) & \ddots & \vdots \\ & & \ddots & f'(\lambda_k) \\ & & & f(\lambda_k) \end{bmatrix}$$

Se \mathbf{A} è diagonalizzabile, allora $\mathbf{A} = \mathbf{Z}\mathbf{J}\mathbf{Z}^{-1}$ è la decomposizione spettrale della matrice \mathbf{A} , infatti $\mathbf{J} = \text{diag}(\lambda_i)$ e \mathbf{Z} contengono gli autovalori e gli autovettori corrispondenti di \mathbf{A} rispettivamente. Dalla 1.23 si ha:

$$f(\mathbf{A}) =: \mathbf{Z}f(\mathbf{J})\mathbf{Z}^{-1} = \mathbf{Z}\text{diag}(f(\lambda_i))\mathbf{Z}^{-1}.$$

Se poniamo ω la misura nell'intervallo $[a, b]$ e f una funzione il cui integrale di Stieltjes esiste per tutti i momenti.

Definizione 1.7.2: Formula di quadratura

Una regola di quadratura è la relazione

$$\int_a^b f(\lambda)d\omega = \sum_{j=1}^N w_j f(t_j) + R[f].$$

I valori di t_j e w_j sono i nodi e i pesi della regola di quadratura rispettivamente. La regola ha precisione algebrica d se $R[p] = 0$ per tutti i polinomi p di grado d e non ci sono polinomi q di grado $d + 1$ per i quali $R[q] = 0$.

Approssimando la funzione f con un polinomio interpolante, possiamo ottenere la regola di quadratura di grado $N - 1$. Possiamo considerare l'approssimazione dell'integrale di Stieltjes con la regola di quadratura di Gauss. La caratteristica di queste formule è che i nodi sono le radici del polinomio ortogonale p_N dato dalla formula ricorsiva dei tre termini 1.16 e i pesi sono calcolati in modo da avere una formula interpolatoria.

Definizione 1.7.3:

La formula generale di una regola di quadratura di Gauss è data da

$$I(f) = \int_a^b f(\lambda)d\omega(\lambda) = \sum_{j=1}^N w_j f(t_j) + \sum_{k=1}^M v_k f(z_k) + R[f], \quad (1.24)$$

dove i pesi $[w_j]_{j=1}^N$, $[v_k]_{k=1}^M$ e i nodi $[t_j]_{j=1}^N$ possono essere determinati mentre $[z_k]_{k=1}^M$ sono assegnati.

Il termine $R[f]$ non può essere calcolato esplicitamente. Se le misure di ω sono funzioni decrescenti positive e se f ha una forma abbastanza regolare, allora

$$R[f] = \frac{f^{(2N+M)}(\eta)}{(2N+M)!} \int_a^b \prod_{k=1}^M (\lambda - z_k) \left[\prod_{j=1}^N (\lambda - t_j) \right]^2 d\omega(\lambda), \quad a < \eta < b.$$

Se $M = 0$ si ottiene la regola di Gauss. Se $M = 1$ e $z_1 = a$ o $z_1 = b$ si ha la regola di Gauss-Radau.

Regola di Gauss

Teorema 1.7.1:

Prendendo \mathbf{J}_N la sottomatrice principale di ordine N della matrice di Jacobi 1.17. Allora i nodi t_j^G della formula di quadratura di Gauss sono gli autovalori $\theta_i^{(N)}$ di \mathbf{J}_N e i pesi w_j^G sono il quadrato delle prime componenti normalizzate degli autovettori $\omega_i^{(N)}$

Teorema 1.7.2:

Supponendo che f sia tale che $f^{(2n)}(\xi) > 0 \forall n$ e $\forall \xi, a < \xi < b$ e ponendo

$$\mathcal{G}_N f = \sum_{i=1}^N f(\theta_i^{(N)}) \omega_i^{(N)} \quad (1.25)$$

dove $\theta_i^{(N)}$ e $\omega_i^{(N)}$ sono dati dal teorema 1.7.1. Allora la regola di Gauss è esatta per polinomi di grado inferiore o uguale a $2N - 1$ e

$$\mathcal{G}_N f \leq I(f),$$

la regola è un limite inferiore dell'integrale di Stieltjes.

Questo teorema rappresenta la connessione tra la formula di quadratura di Gauss e l'algoritmo di Lanczos presentato 1.6.2.

Teorema 1.7.3:

La regola di Gauss può essere riscritta:

$$\mathcal{G}_N f = \mathbf{e}_1^T f(\mathbf{T}_N) \mathbf{e}_1. \quad (1.26)$$

Dove \mathbf{T}_N è la matrice tridiagonale di Lanczos.

La regola di Gauss-Radau

La regola di Gauss-Radau è data dalla 1.24, quando $M = 1$. Per calcolare la regola occorre estendere la matrice \mathbf{T}_N di Lanczos, in modo da *comprendere* un autovalore. Supponendo che $z_1 = b$ ovvero il limite superiore dell'intervallo di integrazione. Desideriamo costruire un polinomio p_{N+1} tale che $p_{N+1}(b) = 0$. Dalla formula ricorsiva dei tre termini 1.16, abbiamo:

$$0 = \beta_{N+1} p_{N+1}(b) = (b - \hat{\alpha}_{N+1}) p_N(b) - \beta_N p_{N-1}(b)$$

da cui:

$$\hat{\alpha}_{N+1} = b - \beta_N \frac{p_{n-1}(b)}{p_n(b)}$$

con b autovalore. La matrice di Lanczos si “**allarga**”:

$$\hat{T}_{N+1} = \begin{bmatrix} T_N & \beta_N \\ \beta_N & \hat{\alpha}_{N+1} \end{bmatrix}. \quad (1.27)$$

Come per la regola di Gauss i nodi sono gli autovalori e i pesi sono il quadrato delle prime componenti degli autovettori. Il termine resto per la regola di Gauss-Radau è dato da:

$$R[f] = \frac{f^{(2N+1)}(\eta)}{(2N+1)!} \int_a^b (\lambda - z_1) \left[\prod_{j=1}^N (\lambda - t_j) \right]^2 d\omega(\lambda), \quad a < \eta < b.$$

Teorema 1.7.4:

Ponendo f tale che $f^{(2n+1)}(\xi) > 0, \forall n$ e $\forall \xi, a < \xi < b$ e ponendo:

$$\hat{\mathcal{G}}_{(N+1)} f^b = \sum_{i=1}^N f(t_i^b) \omega_i^b + f(b) v_1^b, \quad (1.28)$$

$$\hat{\mathcal{G}}_{(N+1)} f^a = \sum_{i=1}^N f(t_i^a) \omega_i^a + f(a) v_1^a, \quad (1.29)$$

con i nodi e pesi calcolati con b o a pari ad un autovalore.

Allora la formula di Gauss-Radau è esatta per polinomio di grado inferiore o uguale a $2N$

$$\hat{\mathcal{G}}_{(N+1)} f^a \leq I(f) \leq \hat{\mathcal{G}}_{(N+1)} f^b$$

Forme bilineari

Data $\mathbf{A} \in \mathbb{R}^{n \times n}$ possiamo definire una forma bilineare come una relazione del tipo

$$\mathbf{u}^T f(\mathbf{A}) \mathbf{v}, \quad \mathbf{u}, \mathbf{v} \in \mathbb{R}^n. \quad (1.30)$$

Se per semplificare consideriamo la matrice \mathbf{A} simmetrica e ponendo $\mathbf{u} = \mathbf{v}$ con $\|\mathbf{u}\| = 1$. Usando la decomposizione spettrale 1.4, l'espressione può essere scritta come integrale di Stieltjes:

$$\mathbf{u}^T f(\mathbf{A}) \mathbf{u} = \sum_{i=1}^n f(\lambda_i) \omega_i^2 = \int f(t) d\omega(t), \quad (1.31)$$

dove ω è una funzione di distribuzione costante a tratti con **salto** sugli autovalori λ_i di \mathbf{A} . Un limite inferiore per l'integrale può essere calcolato usando la formula di

quadratura di Gauss al k -esimo punto, e come limite superiore il $(k + 1)$ -esimo punto della formula di quadratura di Gauss-Radau. Se il metodo di Lanczos *va in breakdown*, ovvero se β_{k+1} nella 1.21 si annulla allora lo spettro di \mathbf{T}_k è un sottoinsieme dello spettro \mathbf{A} e la regola di Gauss fornisce l'esatto valore nella forma bilineare della 1.30.

La coppia di regole di quadratura: Gauss e Gauss-Radau possono essere applicate per calcolare i limiti inferiori o superiori per l'espressione nella forma della 1.30 per ogni tipo di funzione analitica di involucro convesso le cui derivate hanno segno costante.

Quando invece $u \neq v$ possiamo riscrivere la 1.30, come:

$$u^T f(\mathbf{A})w = \frac{1}{4}((u + w)^T f(\mathbf{A})(u + w) - (u - w)^T f(\mathbf{A})(u - w)). \quad (1.32)$$

e ovviamente $\|u\| = 1, \|v\| = 1$.

Capitolo 2

Concetti generali di teoria delle reti

NELLA teoria delle reti, una rete può essere pensata come una collezione di agenti connessi tra loro. Per poter studiare matematicamente la rete occorre introdurre alcuni concetti sugli insiemi.

Una rete può rappresentare differenti strutture, nelle quali le varie entità sono rappresentate dai *nodi*, le relazioni che intercorrono tra essi sono rappresentate dagli *archi*.

Definizione 2.0.1:

Una rete, G è una coppia (tupla) (V, E) . Le reti sono note anche con il nome di grafi. V è chiamato **insieme dei vertici** di G ; i suoi elementi sono i **vertici** di G noti come nodi.

- $V = \{v_i\}$, l'insieme finito dei nodi;
- $E \subseteq V \otimes V = \{(v_i, v_j)\}$, l'insieme degli archi, che può a sua volta essere:
 - ① riflessivo se $(v, v) \in E$ per tutti v ;
 - ② anti-riflessivo se $(v, v) \notin E$ per tutti v ;
 - ③ simmetrico se $(v_1, v_2) \in E \iff (v_2, v_1) \in E$.

Definizione 2.0.2:

- Se E è simmetrico allora G è una rete **non orientata**.
- Se E è simmetrico e anti-riflessivo la rete non contiene archi duplicati allora G è detto grafo **semplice**.
- Se E è non simmetrico allora G è una rete **orientata (digrafo)**.

Spesso quando analizziamo una rete siamo interessati alle proprietà individuali di archi e vertici. Possiamo dare alcune definizioni:

Definizione 2.0.3:

- $G_s = (V_s, E_s)$ è un sottografo di $G = (V, E)$ se $V_s \subseteq V$ e $E_s \subseteq V_s \otimes V_s \cap E$.
- Un **loop** in una rete è un arco del tipo (v, v) . Una rete semplice, per definizione, non ha loops.
- Supponiamo che $e = (v_1, v_2)$ sia un arco nella rete $G = (V, E)$. v_1 è incidente **a** e , v_2 è incidente **da** e , e v_1 è adiacente a v_2 . Si noti che se la rete è non orientata e v_1 è adiacente a v_2 allora v_2 è adiacente a v_1 .
- Le reti $G_1 = (V_1, E_1)$ e $G_2 = (V_2, E_2)$ sono **isomorfe** se c'è una corrispondenza biunivoca, $f: V_1 \rightarrow V_2$ tale che, per tutti gli $u, v \in V_1$, il numero di archi tra u e v corrisponde al numero di archi tra $f(u)$ e $f(v)$.
- Se $G = (V, E)$ è una rete semplice allora il suo **complemento** $\bar{G} = (V, \bar{E})$ è la rete semplice il cui insieme dei nodi coincide con quello di G e il cui insieme degli archi è costituito dalle coppie $(u, v) \in E \iff (u, v) \notin \bar{E}$.
- Siano $G_1 = (V_1, E_1)$ e $G_2 = (V_2, E_2)$ due reti tali che $V_1 \cap V_2 = \emptyset$. L'**unione** di queste due reti, $G_1 \cup G_2$ ha l'insieme dei nodi $V_1 \cup V_2$ e l'insieme degli archi $E_1 \cup E_2$. Questa unione avrà (almeno) due componenti disgiunte. G_1 e G_2 sono sottografi di $G_1 \cup G_2$.
- Sia $G = (V, E)$ e $F \subseteq E$. Allora $H = (V, E - F)$ è il sottografo che si ottiene rimuovendo gli archi presenti in F da G . Se $\hat{G} = (V, F)$ allora possiamo scrivere $H = G - \hat{G}$.
- Il **grado in uscita (out degree)** di un nodo è il numero di archi che incidono ad esso, il **grado in ingresso (in degree)** il numero di archi che incidono da esso.

Reti e matrici

Per poter rappresentare le proprietà di una rete possiamo etichettare i nodi della rete. Per esempio in una rete con n nodi possiamo etichettare ciascuno di questi, e otterremo l'insieme: $V = \{1, 2, \dots, n\}$.

Definizione 2.1.1: Matrice di adiacenza

Sia $G = (V, E)$ una rete semplice tale che $V = \{1, 2, \dots, n\}$. Per $1 \leq i, j \leq n$ definiamo:

$$a_{ij} = \begin{cases} 1, & \text{se } (i, j) \in E, \\ 0, & \text{se } (i, j) \notin E. \end{cases}$$

Allora la matrice quadrata $\mathbf{A} = (a_{ij})$ è chiamata **matrice di adiacenza** di G .

La matrice di adiacenza da una rappresentazione non ambigua di come è fatta la rete.

Ogni matrice quadrata può essere interpretata come una rete: chiamando a_{ij} i pesi di un arco allora la rete creata da \mathbf{A} è una rete pesata. I pesi sono utili quando si vuole assegnare una gerarchia agli archi della rete.

Possiamo anche dare un'altra rappresentazione della rete in forma matriciale:

Definizione 2.1.2: Matrice di incidenza

Sia $G = (V, E)$ una rete tale che $V = \{1, 2, \dots, n\}$ e $E = \{e_1, e_2, \dots, e_m\}$ con $e_i = (u_i, v_i)$. Per $1 \leq i \leq m$ e $1 \leq j \leq n$ si definisce

$$b_{ij} = \begin{cases} 1, & \text{se } u_i = j, \\ -1, & \text{se } v_i = j, \\ 0, & \text{se altrimenti.} \end{cases}$$

Allora la matrice rettangolare $B = (b_{ij})$ è chiamata **matrice di incidenza** di G . Se e_i è un loop allora si pone $b_{ii} = 1$ e ogni altro elemento della riga è lasciato a 0.

Cammini e percorsi

La rete evidenzia le connessioni dirette tra nodi, ma le connessioni implicite indirette sono altrettanto¹ importanti di quelle dirette. Se i nodi non sono direttamente connessi, ma hanno dei nodi in comune l'informazione che attraversa i nodi può assumere differenti *strade*, presumendo che l'informazione possa passare da un nodo ad un altro.

Definizione 2.2.1:

- 1 Un **cammino** in una rete è una serie di archi (non necessariamente distinti)

$$(u_1, v_1), (u_2, v_2), \dots, (u_p, v_p),$$

dove $v_i = u_{i+1}$ ($i = 1, 2, \dots, p - 1$). Se $v_p = u_1$ allora il cammino è **chiuso**.

- 2 Un **percorso (path)** è un trail nel quale tutte le u_i sono distinte.
- 3 Un percorso chiuso è chiamato **ciclo** o **circuito**.
- 4 Un grafo nel quale non sono presenti cicli è chiamato **aciclico**.
- 5 Un ciclo di lunghezza 3 è chiamato **triangolo**.

Le lunghezze dei cammini, sentieri, percorsi sono date da p .

Possiamo numerare i cammini usando la matrice di adiacenza \mathbf{A} . Le entrate di \mathbf{A}^p ci dicono il numero di cammini di lunghezza p tra ciascuna coppia di nodi.

¹Se non più.

Capitolo 3

Multilayer Network

UNA *multilayer network*¹ è una rete che modella vari tipi di interazioni tra tipi diversi di nodi.

Per esempio se c'è un arco tra una coppia di nodi, allora questi nodi sono *adiacenti* tra loro. L'arco è *incidente* a ciascuno dei nodi, se due archi sono incidenti allo stesso nodo allora si dice che sono incidenti tra loro.

La diversità rispetto alle reti monostrato viene rappresentata dalle strutture dette *layer*. Per rappresentare una multilayer network, con differenti livelli o con differenti tipi di archi, dobbiamo considerare, nella sua rappresentazione, una struttura che abbia oltre ai nodi ed archi dei *layers*.

In una struttura multilayer i nodi appartengono ad ogni strato e gli archi connettono i nodi tra loro nello strato, ma possono anche connettere due nodi appartenenti a strati differenti. Per esempio, un nodo u in uno strato α , può connettere un nodo v in uno strato β , come nell'esempio in figura 3.1.

Può essere definito come *aspetto*² della multilayer network, una dimensione³ della rete *caratterizzante* della rete, per esempio il tempo, o il tipo di interazione etc. Ad un aspetto può essere associato il concetto di dimensione, in matematica la parola dimensione ha un preciso significato, ma qui nelle reti, alla parola dimensione, corrisponde uno strato nella multilayer network.

I primi due elementi della multilayer network M producono un grafo: $G_M = (V_M, E_M)$, dunque una multilayer network può essere interpretata come un grafo nel quale i nodi sono etichettati in un certo modo, si veda la figura 3.1. Questa osservazione rende più facile la generalizzazione di molti concetti base. Si potrebbe definire un *peso* per un multilayer network M definendo il peso degli archi del grafo sottostante G_M rappresentando gli archi con un numero reale sfruttando la funzione $w: E_M \rightarrow \mathbb{R}$. Inoltre la multilayer network M può essere **non diretta (diretta)** se il grafo sottostante G_M è non diretto (diretto).

Per una notazione più compatta, gli strati elementari possono essere rappresentati in grassetto infatti: $(u, \alpha_1, \dots, \alpha_d) \equiv (u, \alpha)$. Sfruttando questa notazione, una multi-

¹ Si riportano i nomi *originali*, evitando la traduzione.

² Anche la parola *caratteristica* sarebbe ragionevole.

³ Dove per dimensione si intende uno strato della rete.

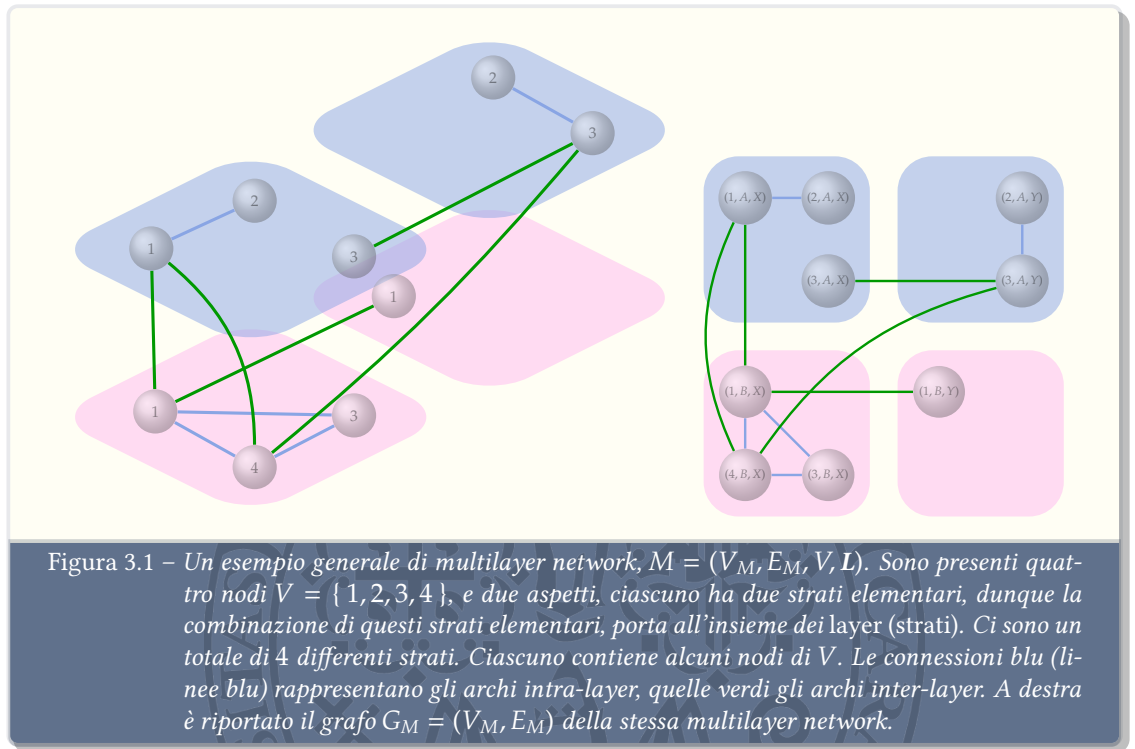


Figura 3.1 – Un esempio generale di multilayer network, $M = (V_M, E_M, V, L)$. Sono presenti quattro nodi $V = \{1, 2, 3, 4\}$, e due aspetti, ciascuno ha due strati elementari, dunque la combinazione di questi strati elementari, porta all'insieme dei layer (strati). Ci sono un totale di 4 differenti strati. Ciascuno contiene alcuni nodi di V . Le connessioni blu (linee blu) rappresentano gli archi intra-layer, quelle verdi gli archi inter-layer. A destra è riportato il grafo $G_M = (V_M, E_M)$ della stessa multilayer network.

layer network è non direzionale (non diretta, indiretta) se: $((u, \alpha), (v, \beta)) \in E_M \Rightarrow ((v, \beta), (u, \alpha)) \in E_M$.

In una prima analisi⁴ non vengono presi in considerazione corto-circuiti, o loop, ovvero non esistono archi del tipo: $((u, \alpha), (u, \alpha)) \notin E_M$.

È spesso conveniente utilizzare una semantica diversa per gli archi presenti in uno strato e per gli archi che attraversano lo strato. È utile dividere l'insieme degli archi come:

- 1 $E_A = \{((u, \alpha), (v, \beta)) \in E_M \mid \alpha = \beta\}$ l'insieme di archi intra-strato⁵;
- 2 $E_C = E_M \setminus E_A$ l'insieme di archi inter-strato⁶.

come in figura 3.1.

Risulta anche utile definire l'insieme degli archi accoppiati⁷: $E_{\bar{C}} \subseteq E_C$ come archi per il quale due nodi rappresentano la stessa entità in strati differenti: $E_{\bar{C}} = \{((u, \alpha), (u, \beta)) \in E_C\}$.

Si possono definire anche i seguenti grafi, figura 3.2:

- 1 $G_A = (V_M, E_A)$ grafo intra-strato;

⁴Senza ledere in generalità.

⁵In sostanza gli archi tra nodi differenti nello stesso strato.

⁶Gli archi tra nodi in differenti strati.

⁷Coupling edge.

- 2 $G_C = (V_M, E_C)$ grafo inter-strato;
- 3 $G_{\tilde{C}} = (V_M, E_{\tilde{C}})$ grafo accoppiato.

Classificazione reti multistrato

Sulla base dei vincoli esterni, una multilayer network può essere classificata come:

- 1 *node-aligned* (totalmente interconnessa) se tutti gli strati contengono tutti i nodi: $V_M = V \times L_1 \times \dots \times L_d$, diremo inoltre che il numero di entità, (che è rappresentato dai nodi) per questo tipo di reti coincide con: $n = |V|$;
- 2 *layer-disjoint* se ciascun nodo esiste in al più uno strato $(u, \alpha), (u, \beta) \in V_M \Rightarrow \alpha = \beta$;
- 3 *diagonal* se tutti gli archi inter-strato sono tra i nodi e le loro corrispettive controparti: $E_{\tilde{C}} = E_C$.

Diremo che una multilayer network diagonale è *layer-coupled* se gli archi di accoppiamento e i loro pesi sono indipendenti dai nodi:

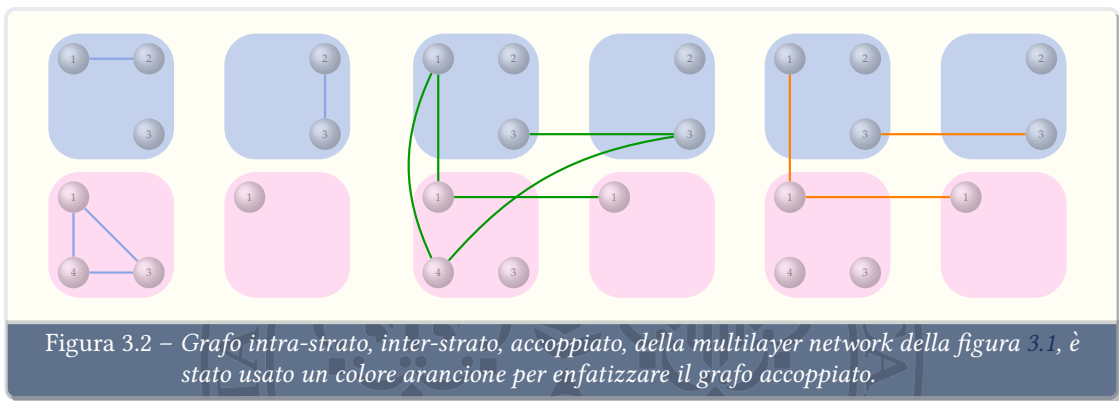
$$\begin{aligned} ((u, \alpha), (u, \beta)) \in E_C \text{ e } ((v, \alpha), (v, \beta)) \in V_M \Rightarrow ((v, \alpha), (v, \beta)) \in E_C \\ \text{ e } w((u, \alpha), (u, \beta)) = w((v, \alpha), (v, \beta)) \quad \forall u, v, \alpha, \beta. \end{aligned}$$

Questo vuol dire che presi due strati, l'accoppiamento è lo stesso per tutti i nodi.

L'accoppiamento è *categorical* se ciascun nodo è adiacente a tutte le sue controparti in altri strati:

$$((u, \alpha), (u, \beta)) \in V_M \Rightarrow ((u, \alpha), (u, \beta)) \in E_M.$$

Quando $d = 1$ è presente un singolo aspetto e l'insieme degli strati risulta: $L = L_1$, il numero degli strati è pari a $b = |L_1|$. Si noti come una multilayer network categoricamente-accoppiata è un sottoinsieme di una rete strato-accoppiata, che a loro volta sono un sottoinsieme di una rete diagonale.



Rappresentazione tensoriale

Per descrivere la multilayer network nodo-allineata⁸ usiamo il tensore di adiacenza a rango $2(d + 1)$:

$$\mathcal{A} \in \{0, 1\}^{|V| \times |V| \times |L_1| \times |L_1| \times \dots \times |L_d| \times |L_d|}$$

che in forma compatta può essere espressa come:

$$\mathcal{A}_{uv\alpha\beta} = \mathcal{A}_{uv\alpha_1\beta_1 \dots \alpha_d\beta_d}$$

l'elemento tensoriale $\mathcal{A}_{uv\alpha\beta}$ assume il valore 1 se e soltanto se $((u, \alpha), (v, \beta)) \in E_M$, 0 altrimenti.

Può essere definito un tensore di adiacenza pesato⁹ \mathcal{W} nel quale il valore per ciascun elemento corrisponde al peso di un arco, 0 se l'arco non c'è.

Questa rappresentazione tensoriale benché sia appropriata a descrivere una multilayer network nodo-allineata può essere impiegata per altri tipi di multilayer network¹⁰, impiegando nodi estranei: *nodi vuoti*, che non sono adiacenti ad altri nodi. Questa tecnica, di *imbottitura*: inserire dei nodi che in realtà non sono presenti, è comoda, ma occorre tenerne conto da un punto di vista matematico, occorre porre attenzione ai risultati che si ottengono.

Facendo l'ipotesi che la multilayer network sia nodo-allineata¹¹, i vincoli forzano molti elementi del tensore ad essere 0, questo permette di abbassarne il rango.

Se la multilayer network è ad accoppiamento diagonale¹², allora il tensore di adiacenza ha elementi $\mathcal{A}_{uv\alpha\beta} = 0$ se gli indici dei due nodi differiscono tra loro: $u \neq v$ e gli strati non sono gli stessi $\alpha \neq \beta$. Con questi vincoli la multilayer network può essere espressa come combinazione di un tensore di adiacenza intra-strato con elementi $\mathcal{A}_{uv\alpha} = \mathcal{A}_{uv\alpha\alpha}$ e un tensore di accoppiamento con elementi $C_{u\alpha\beta} = \mathcal{A}_{uu\alpha\beta}$ ¹³.

Se in una multilayer network diagonale non fosse permesso un accoppiamento inter-aspetto¹⁴, il tensore di adiacenza avrebbe elementi nulli quando gli indici dello strato differiscono l'uno dall'altro per al più un aspetto: $\mathcal{A}_{uv\alpha\beta} = 0$ se entrambi: $u \neq v$ e $\alpha \neq \beta$ o se $u = v$, ma α e β differiscono al più di un elemento¹⁵.

Appiattimento del tensore

Si può ridurre il numero degli aspetti, in un tensore di adiacenza combinando l'aspetto i e l'aspetto j in un nuovo aspetto h . Si passa da una multilayer network nodo-allineata $M = (V_M, E_M, V, \mathbf{L})$ con d aspetti, a una nuova multilayer network $M' = (V_{M'}, E_{M'}, V, \mathbf{L}')$ con $d - 1$ aspetti, il numero totale degli elementi, nel tensore corrispondente, rimane lo stesso: $L'_h = L_i \times L_j$, così $|L'_h| = |L_i||L_j|$.

⁸Dove sono presenti tutti i nodi in ciascuno strato.

⁹Così come si definiva una matrice di adiacenza pesata per un grafo.

¹⁰Dove non sono presenti tutti i nodi.

¹¹Tutti i nodi sono presenti in ogni strato, se così non fosse si procede introducendone dei fittizi.

¹²Ovvero se gli archi inter-strato sono permessi solo tra due rappresentazioni dello stesso nodo.

¹³Nella letteratura questo equivale a una formulazione *multislice*.

¹⁴Si possono avere tanti strati ciascuno dei quali può aver differenti aspetti.

¹⁵Si ricordi che **un** elemento di *boldsymbolalpha* corrisponde ad **un** aspetto.

Esiste una relazione biunivoca tra gli elementi del vecchio tensore e del nuovo.

Questo processo viene chiamato *Flattening*. Partendo da un tensore di adiacenza con un numero di aspetti pari a d , si passa a un tensore di adiacenza con un numero di aspetti pari a $d - 1$, continuando questo processo si può arrivare a un numero di aspetti pari a 1 portando ad una matrice di **supra-adjacency**.

Rappresentazione di supra-adjacency

La rappresentazione matriciale per una multilayer network è estremamente utile e potente, perché sfrutta tutti i *tools*, i metodi e i risultati teorici sviluppati per le matrici. Per poter sfruttare questi tools, occorre riportare la multilayer network a una matrice di sopra-adiacenza. Come riportato nell'esempio della figura 3.1, la matrice di adiacenza di un *supra-graph* corrisponde alla matrice di sopra adiacenza di una multilayer network.

Matematicamente è più semplice trattare matrici che tensori, ed un ulteriore vantaggio della matrice di sopra-adiacenza rispetto al tensore di adiacenza è che se la multilayer network non è nodo allineata¹⁶ non occorre *inserire* nodi vuoti. Per contro questo vantaggio, di ottenere una matrice di sopra-adiacenza, porta ad una *perdita* di informazione su alcuni aspetti. Ma partizionando gli archi della rete in insiemi:

- ① intra-archi;
- ② inter-archi;
- ③ archi di accoppiamento;

è possibile riottenere queste informazioni.

Si possono definire diverse matrici di sopra-adiacenza corrispondenti a i corrispettivi grafi:

- ① matrice sopra-adiacenza $\Rightarrow G_M$;
- ② matrice sopra-adiacenza intra-strato $\Rightarrow G_A$;
- ③ matrice sopra-adiacenza inter-strato $\Rightarrow G_C$;
- ④ matrice sopra-adiacenza di accoppiamento $\Rightarrow G_{\bar{C}}$.

Reti multiple e reti multi relazionali

Una rete *multiplex* e una rete *multirelational* sono simili a reti multistrato e sono come una sequenza di grafi:

$$\{ G_\alpha \}_{\alpha=1}^b = \{ (V_\alpha, E_\alpha)_{\alpha=1}^b \}$$

dove $E_\alpha \subset V_\alpha \times V_\alpha$ è l'insieme degli archi e α è l'indice dei grafi.

¹⁶Non sono presenti tutti i nodi per ogni strato.

Sovente l'insieme dei nodi è lo stesso per differenti strati: $V_\alpha = V_\beta$ per tutti gli α, β o perlomeno condividono molti nodi: $\bigcap_{\alpha=1}^b V_\alpha \neq \emptyset$.

Una rete multiplex potrebbe essere definita come un *edge-colored multigraphs*, reti con tipi multipli di archi.

La definizione di arco-colorato multi grafo può essere data come:

$$G_e = (V, E, C)$$

dove V è l'insieme dei nodi, C l'insieme dei colori (usato per etichettare gli archi) e $E \subset V \times V \times C$ è l'insieme degli archi.

Usando la parola *colore* come etichetta, né segue che archi incidenti allo stesso nodo avranno lo stesso colore.

Gli archi-colorati multi grafo possono rappresentare un insieme di reti multiple, che hanno lo stesso insieme di nodi in ciascuno strato al quale si associa un unico colore.

Un tipo di accoppiamento ben noto è l'accoppiamento *ordinale* nel quale gli strati sono ordinati e i nodi sono adiacenti soltanto alla loro controparte in strati consecutivi.

Un esempio di rete multiplex potrebbe essere una rete di trasposto, dove le città sono adiacenti e gli archi possono essere: la rete ferroviaria, linee aeree, e linee navali. Può essere usata una rete nodo-colorata, i colori rappresentano le tre differenti modalità di trasporto e i nodi rappresentano le stazioni dei treni, gli aeroporti e i porti.

Nella rappresentazione multiplex della stessa rete i nodi sono le città e i tre strati sono rappresentati dai tre modi di trasporto.

Entrambe le due rappresentazioni sono reti multistrato, sono simili, ma non equivalenti totalmente.

La differenza è che la rappresentazione multiplex l'insieme dei nodi è lo stesso per ciascuno strato, mentre la rappresentazione nodo-colorata ha un insieme di nodi disgiunto. In questo tipo di rete nodo-colorata ciascun percorso (path)¹⁷ è rappresentato da un differente colore.

¹⁷Si definisce *cammino* come sequenza di nodi adiacenti, come *percorso* il cammino dove **non** si ripetono i nodi, con l'eccezione che il nodo di partenza e il nodo di arrivo possono essere gli stessi. Un *cammino intra-strato*, si verifica solo all'interno di uno strato, un *percorso intra-strato* lo stesso, si veda 2.2 a pagina 30.

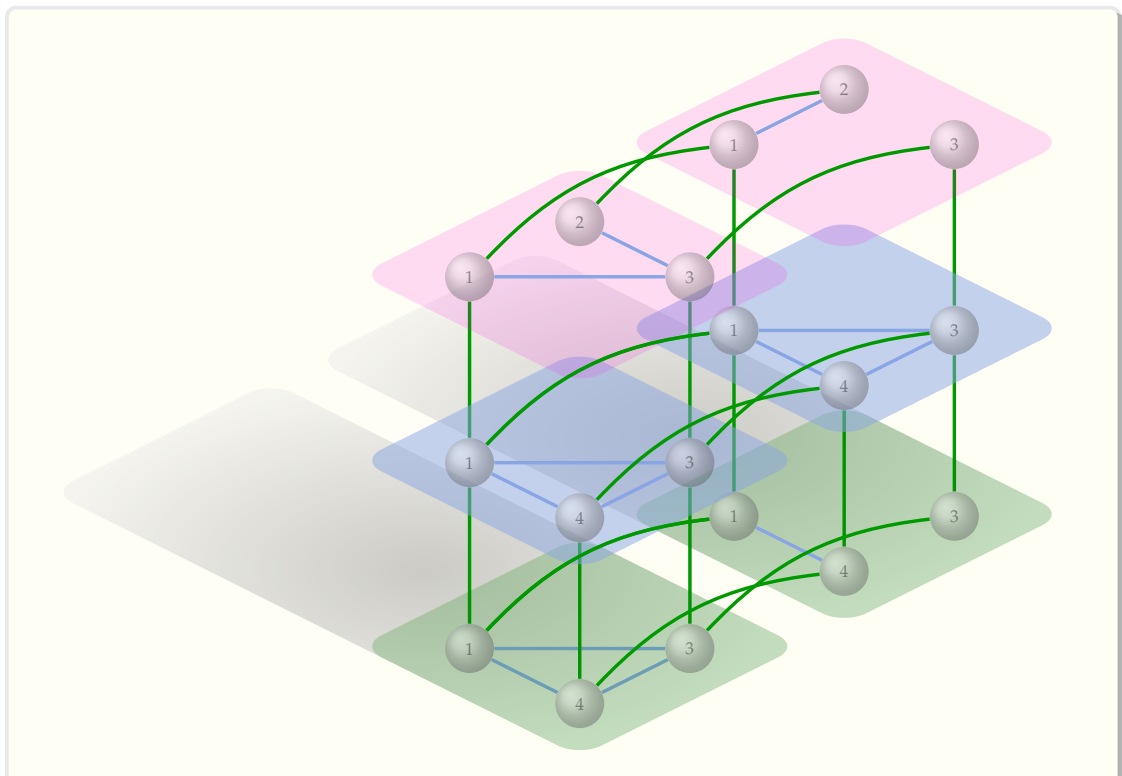


Figura 3.3 – *multilayer network*, si sono presi presi archi indirezionali. Ci sono due aspetti. Sono stati usati colori differenti per rappresentare gli archi intra-strato (blu), e inter-strato (verde). Tutti gli archi sono archi accoppiati perché i nodi sono adiacenti soltanto a se stessi, e a nessun altro nodo. Gli archi di accoppiamento nel primo aspetto sono categorici, perché esiste un arco tra i nodi in ciascuno strato, mentre nel secondo aspetto è ordinale, infatti non tutti i nodi sono connessi per differenti strati.

Capitolo 4

Implementazione e prove numeriche

IN questo capitolo si analizza il codice scritto per le simulazioni, si riportano e si commentano i risultati ottenuti, che vengono riportati come grafici. Viene ora riportato il codice della **function**: **LGQuadratura** scritto per fare le prove numeriche. L'applicazione di questo Algoritmo n volte, dove n è la dimensione della matrice **A** permette di calcolare la **subgraph centrality** per ogni nodo. i -esimo elemento diagonale è la subgraph centrality dell' i -esimo nodo. Il modo *standard* di calcolarla è estrarre la diagonale della **function**: *expm*(\cdot) di Matlab applicata alla matrice di adiacenza **A**. Quando in Matlab si scrive: *expm*(**A**), Matlab applica l'approssimazione di Padé, questo calcolo non sempre è possibile, infatti ci sono due casi in cui si ottiene *overflow*:

- 1 la matrice **A** è di *grande* dimensione: n molto alto;
- 2 la matrice può anche essere piccola: 2×2 , ma valori molto elevati, già per valori prossimi a 10^3 *expm*(**A**) produce *overflow*.

Per questo si vogliono estrarre delle informazioni dalla rete, senza dover calcolare *expm*(**A**). L'algoritmo di seguito presentato per il calcolo della subgraph centrality¹ mette a confronto i risultati ottenuti con i due metodi quello *standard*: calcolo diretto della *expm*(**A**), con quello dell'Algoritmo di Lanczos/Quadratura. Questo quando le dimensioni della matrice di adiacenza **A** lo *permettono*. Si sono svolte diverse prove, a cominciare da matrici di piccole dimensioni: $n = 10$. I grafici riportati nelle figure 4.1, e 4.4, mettono a confronto i tempi di calcolo per i due algoritmi nella prima figura, mentre nella seconda figura si riportano gli errori, ovvero la differenza tra il valore dato dall'approssimazione di Padé e l'algoritmo iterativo di Lanczos/Quadratura. I tempi di calcolo dell'approssimazione di Padé è costante infatti viene svolta *una* sola volta la **function**: *expm*(**A**) e poi viene estratta la diagonale con la **function**: *diag*(\cdot). Mentre con il metodo iterativo, per *ciascun* nodo;

- 1 si costruisce il vettore e_i ;
- 2 si calcola la matrice di Lanczos J_k ;

¹Si veda [Fen13].

3 si esegue $e_i^T \expm(\mathbf{J}_k) e_i$;

4 si memorizza il risultato.

function: LGQuadratura(A, b, n, lambda, varargin)

```
1 normb = norm (b,2);
2 q(:,1) = b/normb;
3 % Calcolo dei primi coefficienti della matrice di Lanczos
4 v = A*q(:,1);
5 T(1,1) = q(:,1)'*v;
6 v = v - T(1,1)*q(:,1);
7 T(1,2) = norm(v,2);
8 % Una verifica che il valore non sia troppo piccolo
9 if(abs (T(1,2)) < eps)
10     risultati = 0; s = 1;
11     breakdown = true;
12     return;
13 else
14     q(:,2) = v / T(1,2);
15 end
16 T(2,1) = T(1,2);
17 G = Gauss (T); % Metodo di Gauss: primo elemento
18 s = 2;
19 breakdown = false; err = inf;
20 risultati = abs(G)*ones(1,3); %risultati(1,2) = Q;
21 tic
22 % Ciclo principale
23 while s<n-1 && breakdown && err > abs(G)*1e-3,
24 % Calcolo degli elementi successivi della matrice
25 % di Lanczos
26     [q, T, breakdown] = SottomatriceLanczos (A, q, T, ...
27                                             s, breakdown);
28     if breakdown,
29         breakdown = false;
30         return;
31     end
32     % Costruzione del polinomio ortogonale
33     P = PolinomioOrtogonale (T, lambda);
34     % Calcolo del valore di Gauss Radau: GR
35     [Tt, GR] = GaussRadau (T, P, lambda);
36     err = abs(G - GR);
37     risultati(s-1,:) = [G, GR, err];
38     G = Gauss(T);
39     s = s+1;
40 end
```

Si misurano i tempi per svolgere queste operazioni, con i comandi di Matlab `tic ... toc`. Gli errori sono calcolati come differenza del valore assoluto tra l'approssimazione di Padé e il metodo iterativo. Queste operazioni sono state fatte per eseguire un confronto ovviamente se le *dimensioni* della rete lo permettono, perché come già detto sopra, se la rete è *abbastanza* grande questo tipo di confronto non può essere eseguito in quanto l'approssimazione di Padé va in overflow.

```
function:      SubgraphCentrality(A,  n,  lambda,
varargin)
```

```
1 tic; SC1 = expm(A); SC1 = diag(SC1); PadeTime = toc;
```

⋮

```
function:      SubgraphCentrality(A,  n,  lambda,
varargin)
```

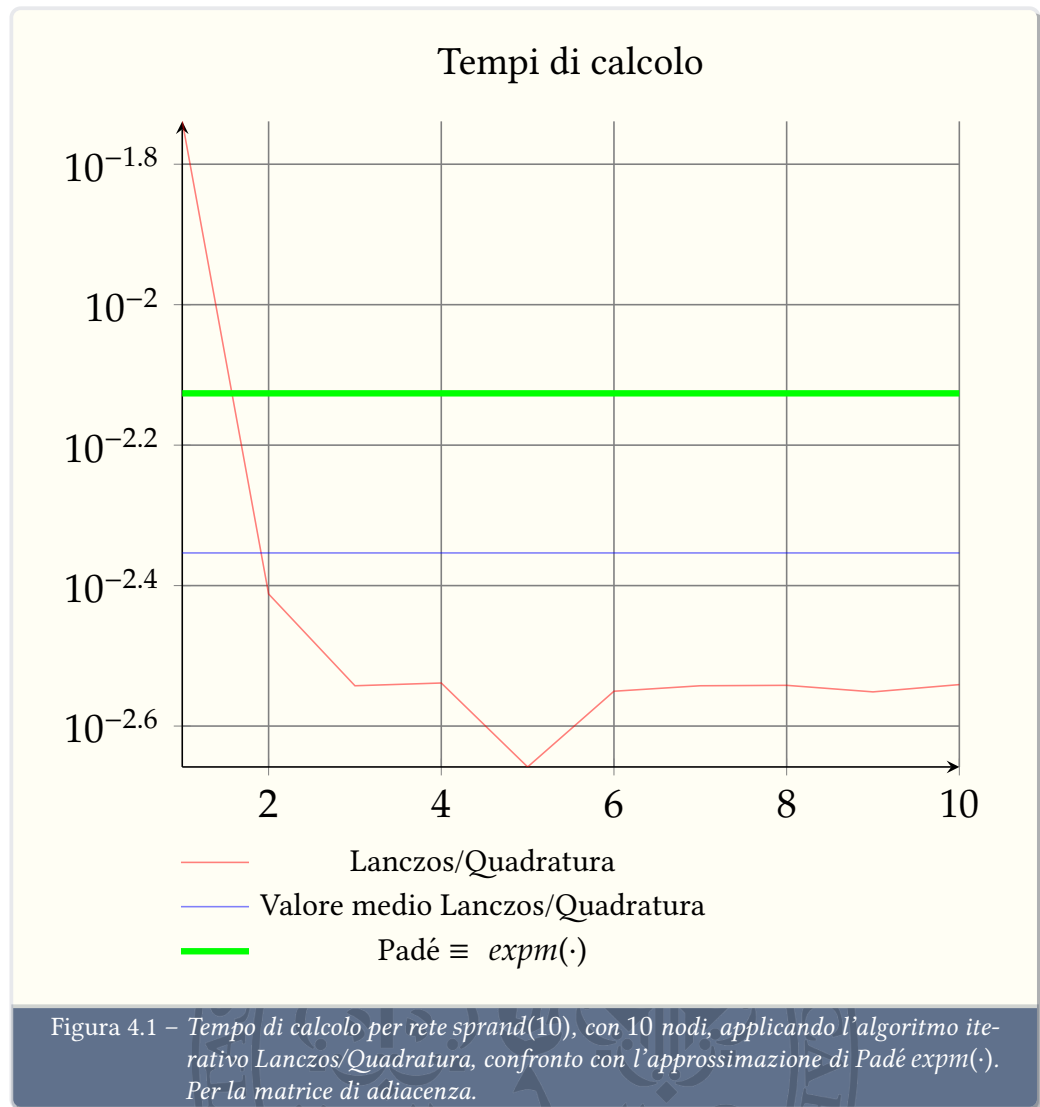
```
1 % Ciclo principale
2 for i = 1:n,
3     e i = zeros(n, 1);
4     e i(i) = 1;
5     tic
6     [ris, s, breakdown] = LGQuadratura(A, e i, n, ...
7                                     lambda, varargin-1);
8     QuadrTime(i) = toc;
9     SC2(i) = ris(end, 1); iteraz(i) = s;
10    Breakdown(i) = breakdown;
11    if rem(i, 10),
12        fprintf('Iterazione: %d\n', i);
13        str = strcat(varargin-2, ' BCK ');
14        Backup(n, i, str, 0, [SC1, SC2]);
15    end
16 end
```

⋮

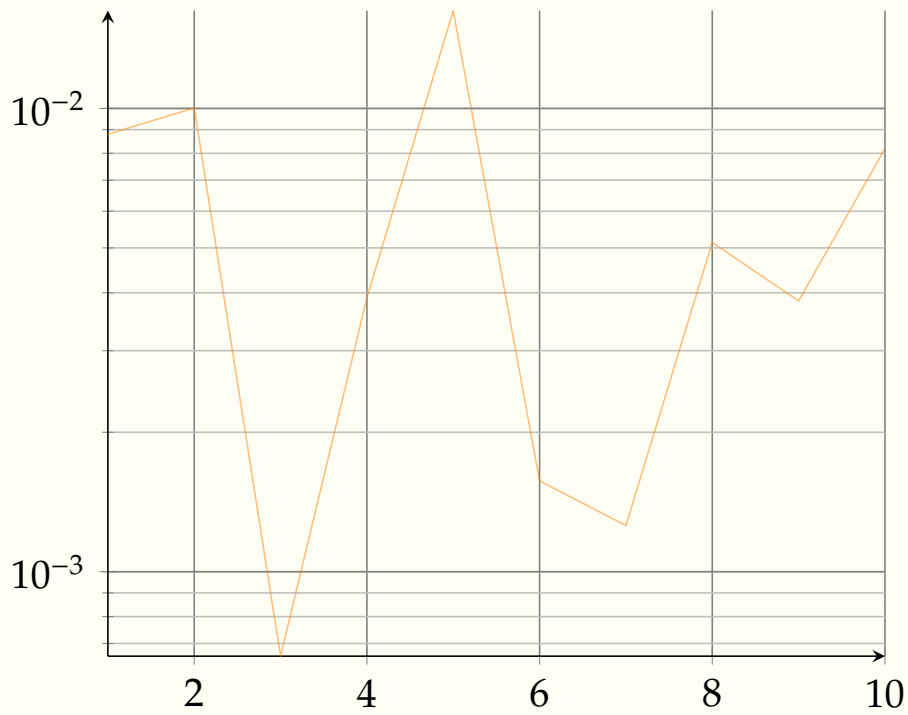
```
function:      SubgraphCentrality(A,  n,  lambda,
varargin)
```

```
1 % Backup finale dei risultati ottenuti
2 Backup(n, i, varargin-2, 1, risultati);
```

Risultati ottenuti



Modulo dell'errore



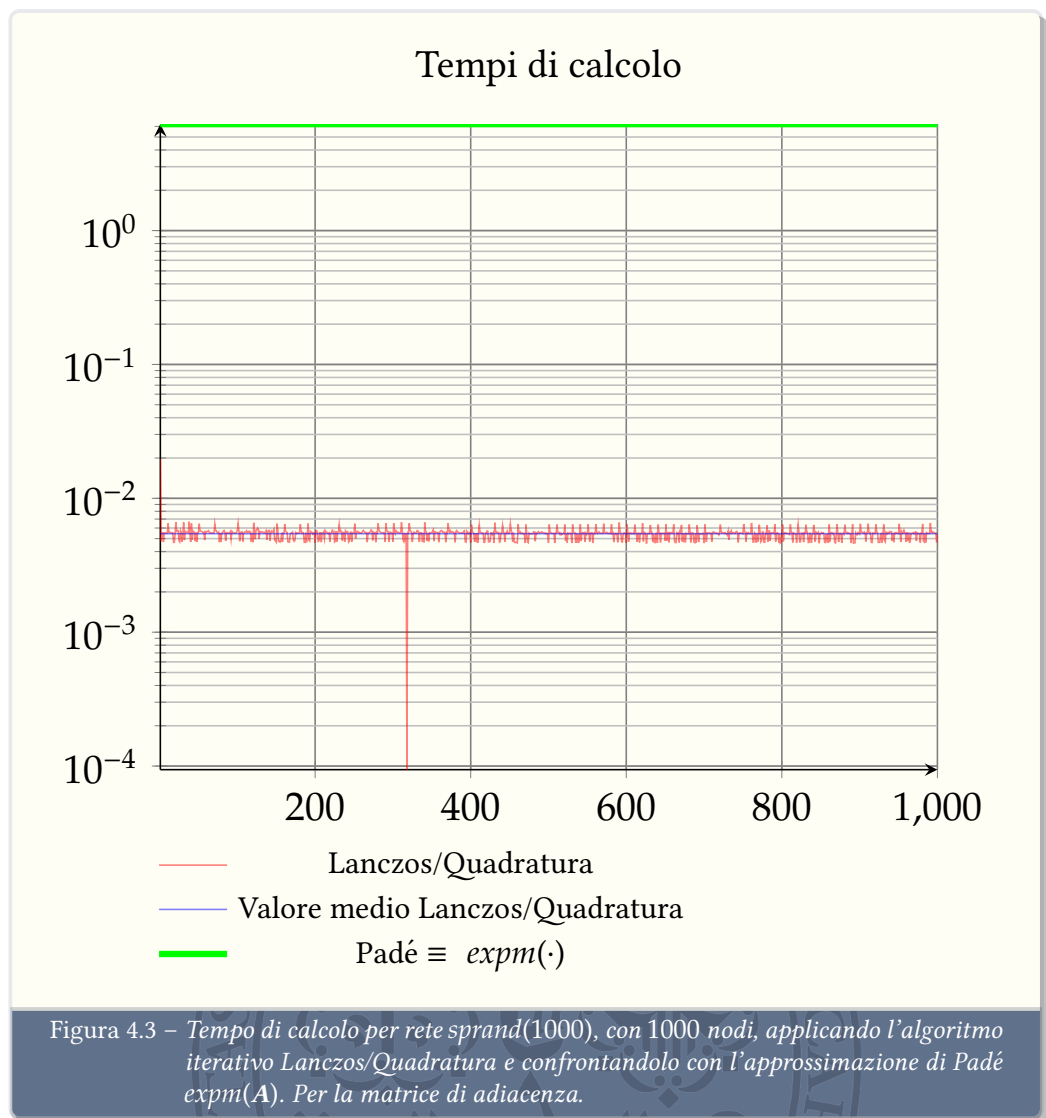
— Errore = $|e_i^T f(\text{Lanczos})e_i - e_i^T f(A)e_i|$.

Figura 4.2 - Calcolo dell'errore come: $|e_i^T f(\text{Lanczos})e_i - e_i^T f(A)e_i|$. Come modulo della differenza tra il valore di Gauss e il valore: $\text{diag}(\text{expm}(A))$. Rete spran(10) con 10 nodi.

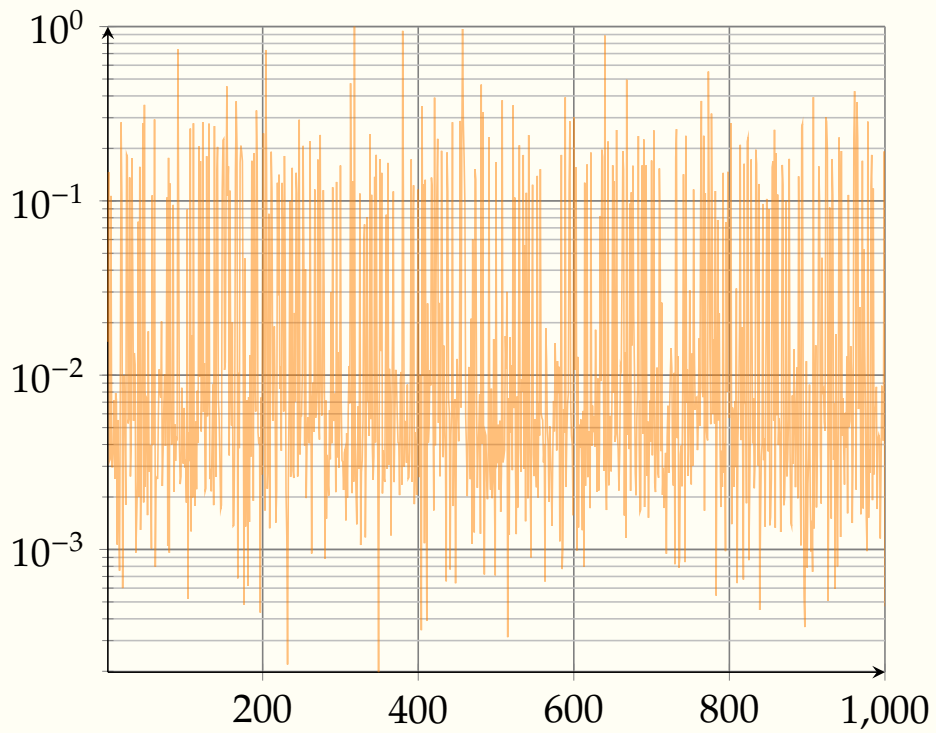
Prove un pò più complesse

Le stesse prove eseguite con matrici un pò più grandi: $n = 1000$ e $n = 2000$ forniscono i risultati riportati nelle seguenti figure.

Prova $n = 1000$



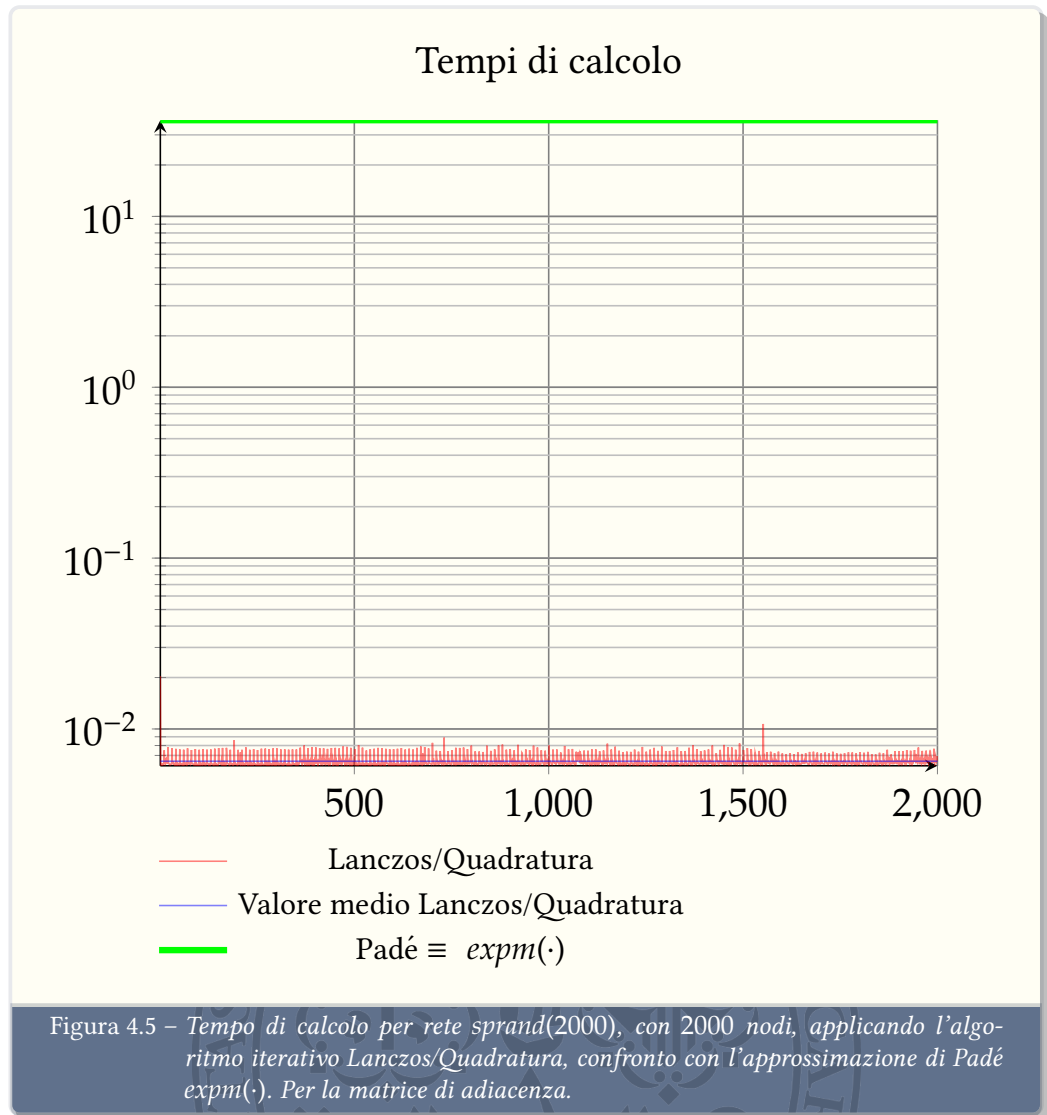
Modulo dell'errore



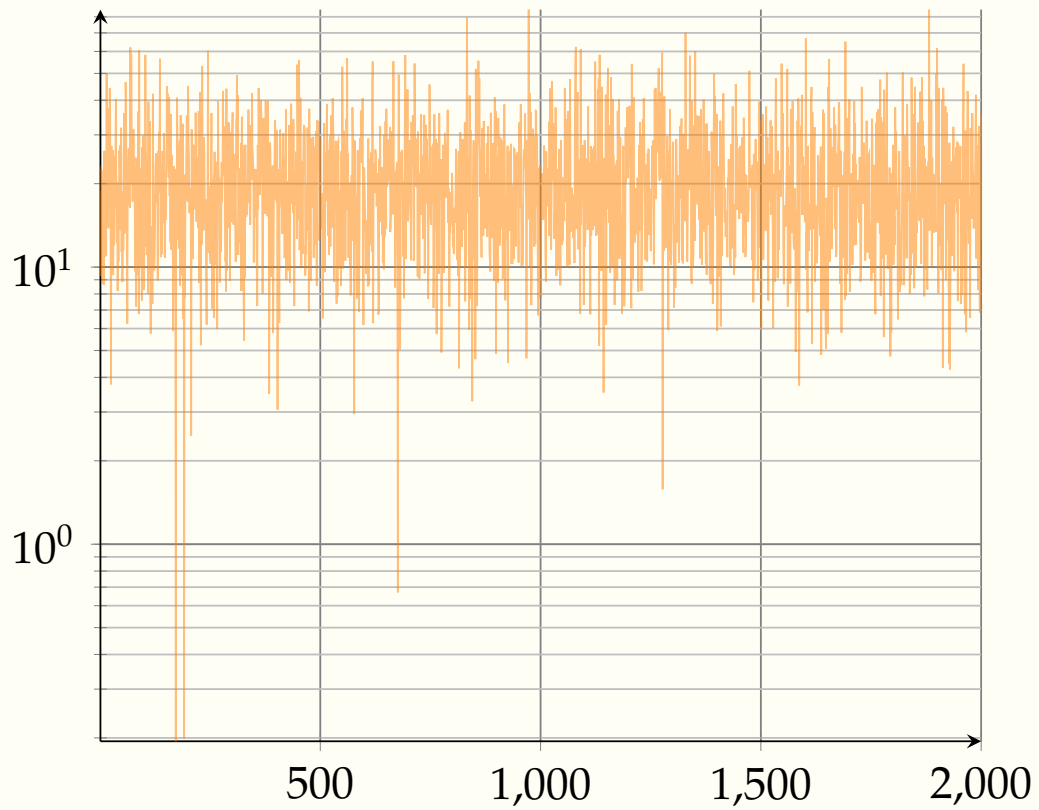
— Errore = $|e_i^T f(\text{Lanczos})e_i - e_i^T f(A)e_i|$.

Figura 4.4 - Calcolo dell'errore come: $|e_i^T f(\text{Lanczos})e_i - e_i^T f(A)e_i|$. Come modulo della differenza tra il valore di Gauss e il valore: $\text{diag}(\text{expm}(A))$. Rete spran(1000) con 1000 nodi.

Prova $n = 2000$



Modulo dell'errore



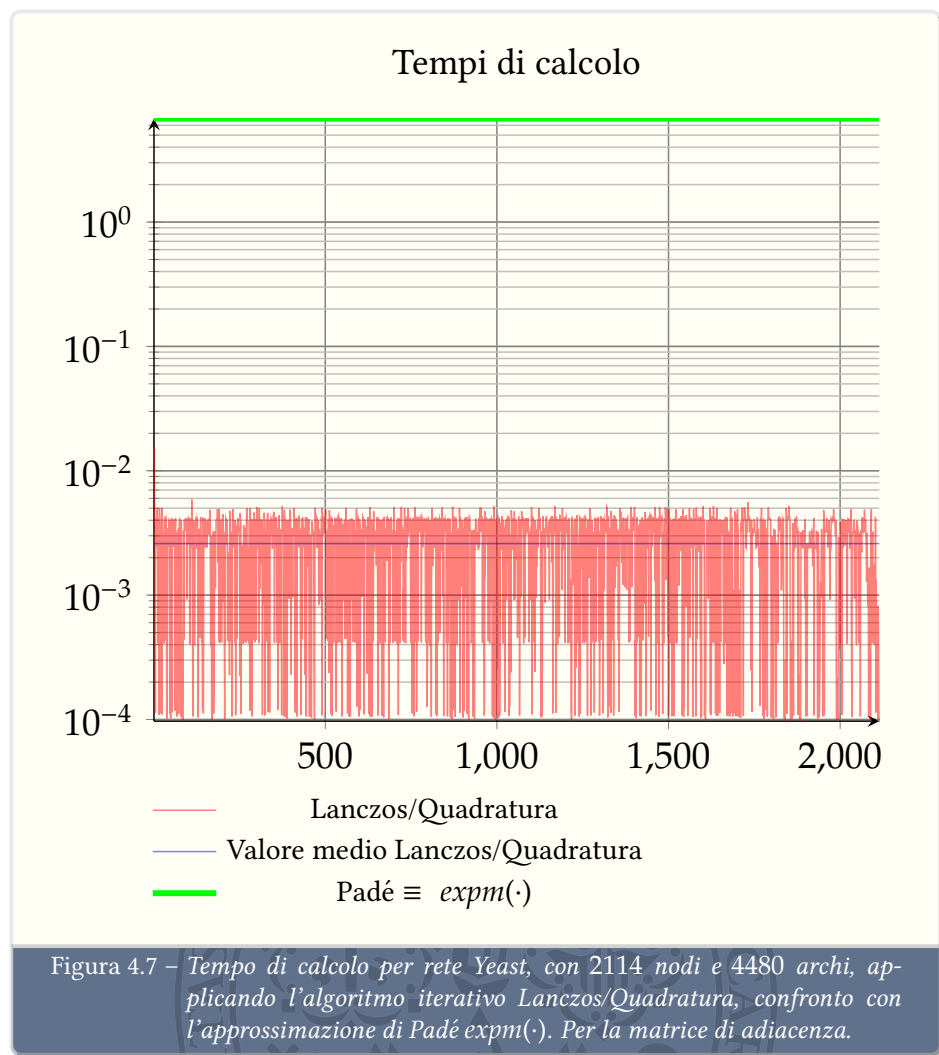
— Errore = $|e_i^T f(\text{Lanczos})e_i - e_i^T f(A)e_i|$.

Figura 4.6 – Calcolo dell'errore come: $|e_i^T f(\text{Lanczos})e_i - e_i^T f(A)e_i|$. Come modulo della differenza tra il valore di Gauss e il valore: $\text{diag}(\text{expm}(A))$. Rete spran(2000) con 2000 nodi.

Prove reali

Si sono svolte delle prove su delle reti che provengono dal *mondo reale*. La prima rete analizzata descrive l'interazione proteica per il lievito: ciascun arco rappresenta l'interazione tra due proteine (2114 nodi, 4480 archi). Per questa rete si riportano tempi ed errori confrontando i due metodi quello iterativo e l'approssimazione di Padé.

Yeast



Modulo dell'errore

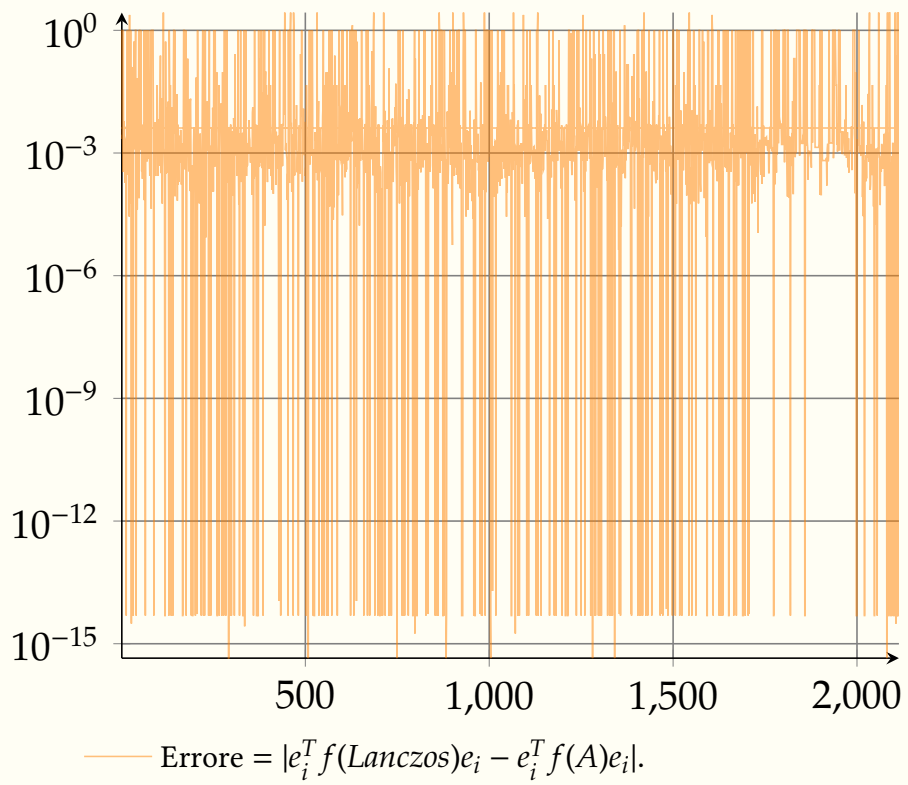
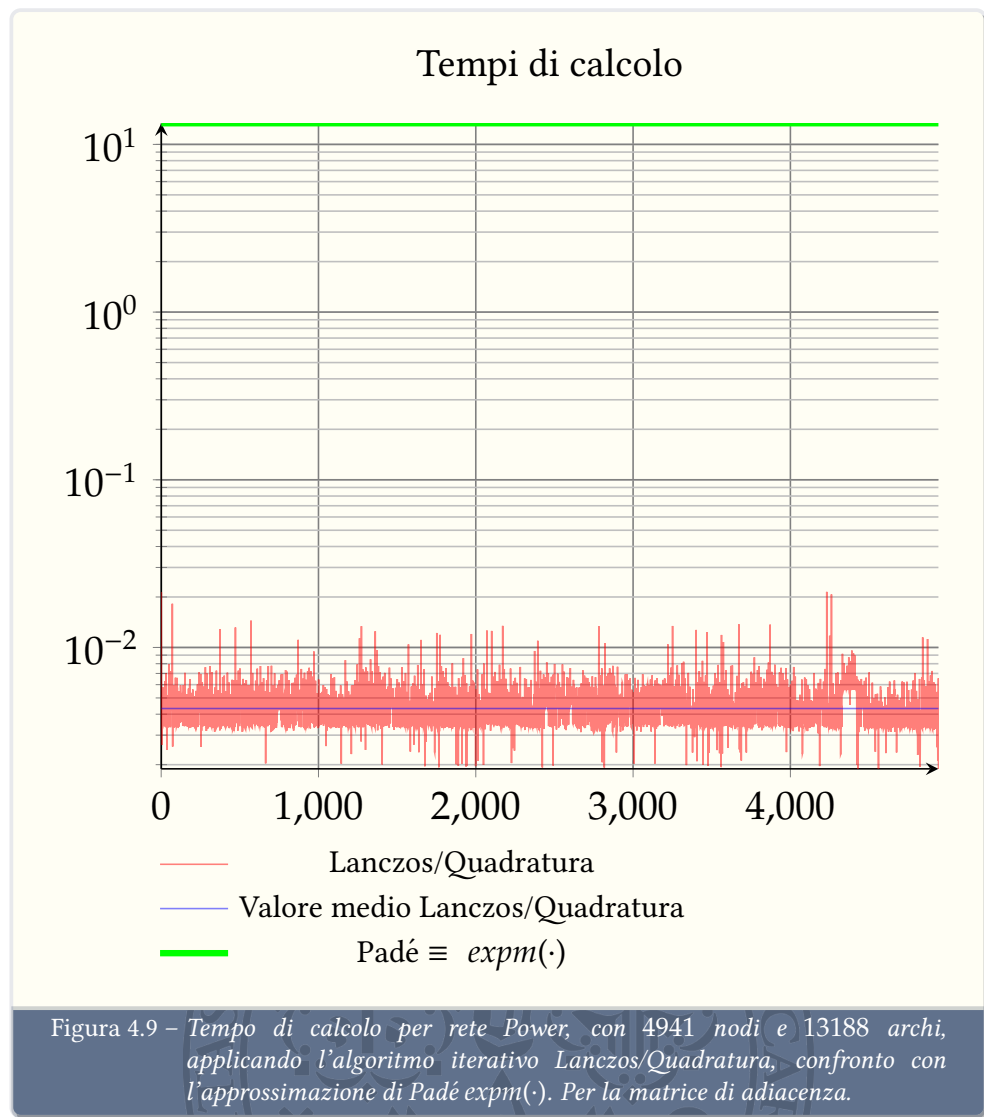


Figura 4.8 - Calcolo dell'errore come: $|e_i^T f(\text{Lanczos})e_i - e_i^T f(A)e_i|$. Come modulo della differenza tra il valore di Gauss e il valore: $\text{diag}(\text{expm}(A))$. Rete Yeast con 2114 nodi e 4480 archi.

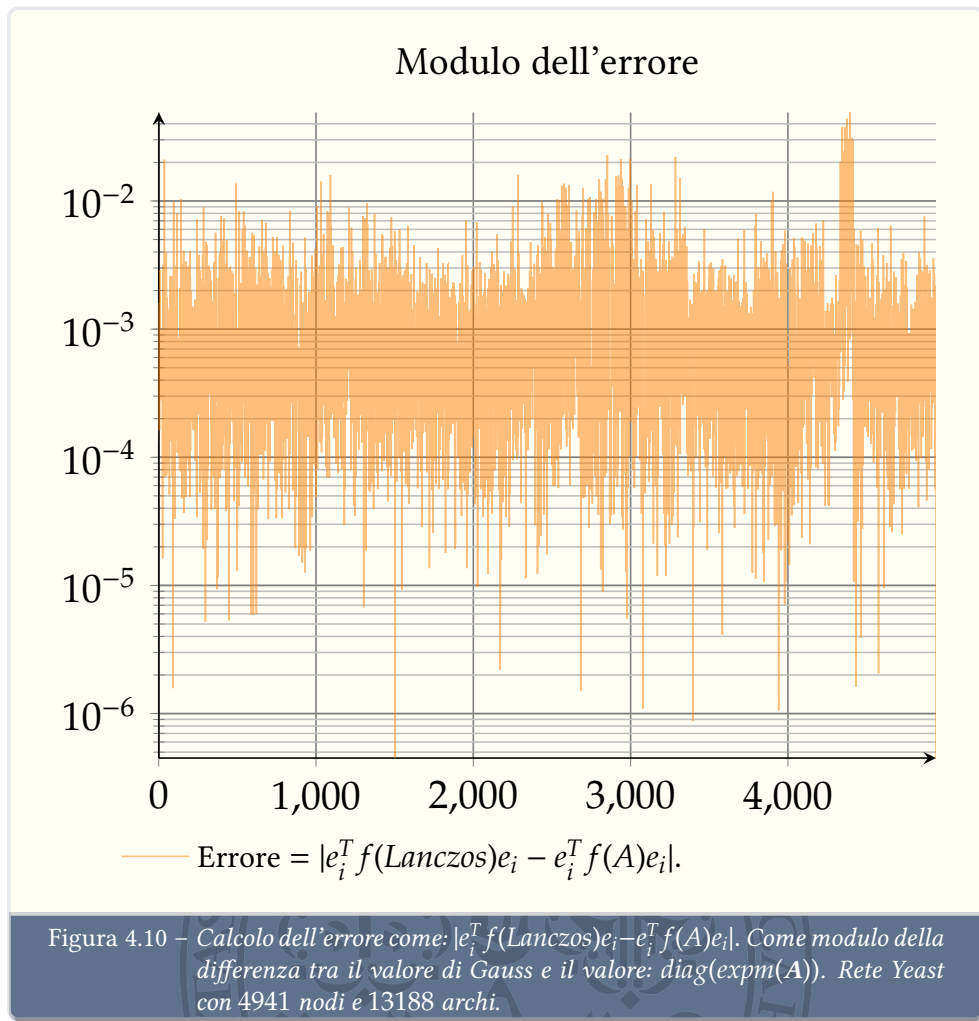
Power

Un'altra rete proveniente dal *mondo reale* rappresenta la rete elettrica degli stati occidentali degli Stati Uniti, (4941 nodi, 13188 archi).



Commenti

Dai grafici ottenuti sui tempi di calcolo, si può notare, figura 4.1 una riduzione dei tempi con il metodo iterativo rispetto al metodo tradizionale: l'approssimazione di



Padé. All'aumentare della dimensione della matrice i tempi di calcolo diventano sempre più differenti, e il metodo iterativo risulta più rapido rispetto all'approssimazione di Padé, figure da 4.3 a 4.9.

Per quanto riguarda i diagrammi degli errori questi variano. Oscillano molto nel caso della rete *yeast*, figura 4.8, oscillano *meno* nel caso della rete *power* figura 4.10.

Facebook

Il più grande esempio proveniente dal *mondo reale* è la rete di amicizie della rete facebook di New Orleans. Ciasun arco tra due utenti vuol dire che il secondo utente appartiene alla lista delle amicizie del primo utente. La rete ha 63731 nodi e 1545686 archi. Le dimensioni di questa rete non permettono il calcolo dell'approssimazione di Padé. Pertanto non può essere eseguito un confronto come fatto nei casi precedenti. Per questa rete si è calcolato e riportato graficamente, la *starting convenience* ovvero la somma delle righe della $\expm(A)$. Dato che questa operazione è impossibile da eseguire direttamente si calcola per ciascun nodo: $e_i^T \expm(A)w$ dove $w = [1 \dots 1]^T$, quindi i due vettori non sono uguali pertanto ricordando la formula polarizzante 1.32 a pagina 26 occorre applicare l'algoritmo di quadratura due volte per ciascun nodo, ciascuno con due vettori laterali uguali. Di seguito si riporta il codice per la simulazione dell'algoritmo.

```
function: StartingConvenience(A, n, lambda,
varargin)

1 % Ciclo principale
2 for i = 1:n,
3     ei = zeros(n, 1);
4     ei(i) = 1;
5     z0 = ei+w;
6     z1 = ei-w;
7     tic
8     [ris0, s0, breakdown0] = LGQuadratura(A, z0, ...
9                                         n, lambda, varargin-1);
10    [ris1, s1, breakdown1] = LGQuadratura(A, z1, ...
11                                         n, lambda, varargin-1);
12    QuadrTime(i) = toc;
13 end
```

⋮

```
function: StartingConvenience(A, n, lambda,
varargin)

1 BackupStartConv(n, i, varargin-2, 1, risultati);
```

Questa simulazione porta al grafico doppiamente logaritmico riportato di seguito:

Starting Convenience rete Facebook 63731 nodi, 1545686 archi.

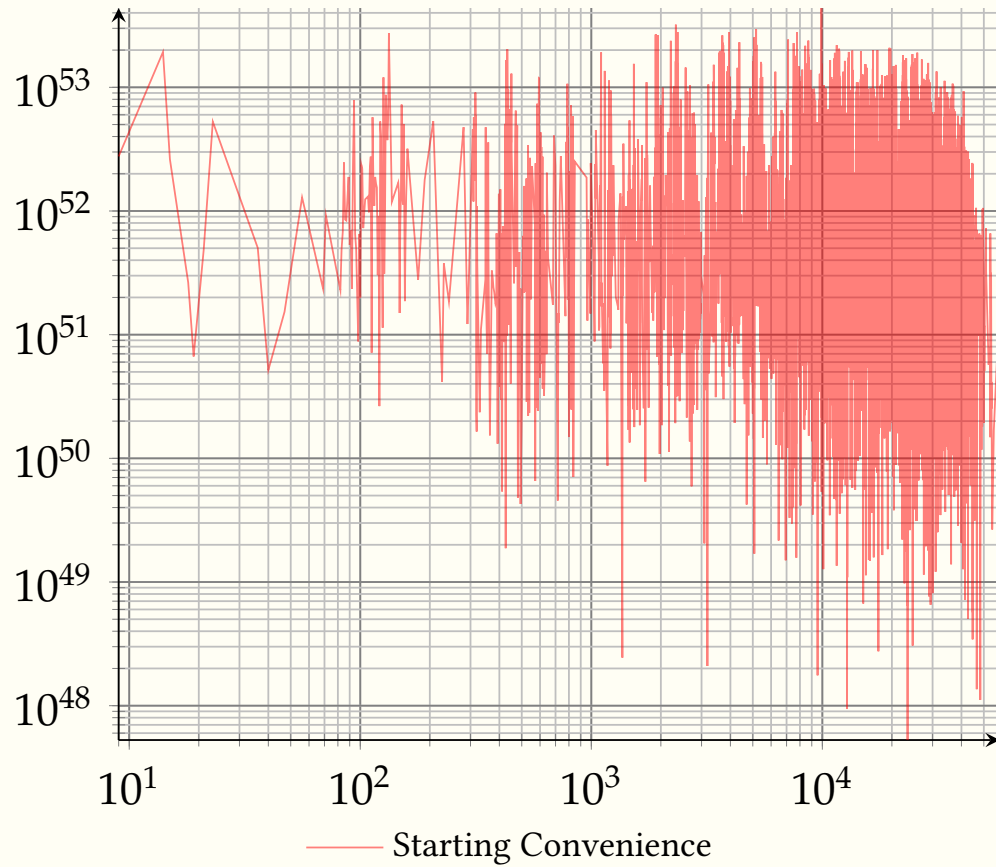


Figura 1 – Starting convenience per la rete Facebook, con 63731 nodi e 1545686 archi. Calcolo della somma delle righe dell'esponenziale $e^T \expm(T) 1$.

Riferimenti bibliografici

- [Fen13] C. Fenu. "Network analysis partial spectral factorization and Gauss quadrature". In: *Siam* (2013).
- [Rod08] Giuseppe Rodriguez. *Algoritmi numerici*. Cagliari: Pitagora Editrice Bologna, 2008.

Indice

- Accoppiamento
 - Categorical, 33
- Algoritmo, iii
- Appiattimento, 34
- Archi, 27
 - Loop, 28
- Arco
 - Incidente, 31
- Autovalore, 5
- Autovettore, 5

- Cammini e percorsi, 30
- Cammino, 30
- Cicli
 - aciclico, 30
- Cili
 - triangolo, 30

- Decomposizione spettrale, 6

- Equivalenza delle norme, 3

- Forma canonica di Jordan, 10
- Formula di quadratura, 23
- Funzione di matrice, 22
- Funzioni matriciali o di matrice, 22

- Grado
 - In degree, 28
 - In ingresso, 28
 - In uscita, 28
 - Out degree, 28

- Insieme archi
 - Anti-riflessivo, 27
 - Riflessivo, 27

- Simmetrica, 27

- Integrale
 - Riemann-Stieltjes, 14
- Iterazioni in sottospazi di Krylov, 17

- Layer, 31

- Matrici
 - adiacenza, 29
 - di forma particolare, 7
 - Hermitiane, 7
 - incidenza, 29
 - Normale, 6
 - Supra-adjacency, 35
 - Unitarie, 7

- Nodi, 27
 - Adiacenti, 31

- Norme
 - Consistente, 9
 - Indotta, 9
 - Matriciale indotta dalla norma 1, 9
 - Matriciale indotta dalla norma 2, 10
 - Matriciale indotta dalla norma vettoriale ∞ , 9
 - Submoltiplicativa, 9

- Numero di condizionamento, 10

- Ortogonalizzazione
 - Gram-Schmidt, 12

- Path, 30
- Percorso, 30
 - Ciclo, 30

- circuito, 30
- Polinomi
 - Matriciali ortogonali, 17
 - Ortogonal, 14
 - Ortonormali, 15
- Problema ben posto, iii
- Prodotto
 - Interno, 14, 16
- Raggio spettrale, 5
- Rango, 4
- Rappresentazione tensoriale, 34
- Rete, 27
 - Complemento, 28
 - Diagonal, 33
 - Edge-colored multigraphs, 36
 - Isomorfe, 28
 - Layer-coupled, 33
 - Layer-disjoint, 33
 - Multilayer, 31
 - Multiplex, 35
 - Multirelational, 35
 - node-aligned, 33
 - Non orientata, 27
 - Orientata, 27
 - Peso, 31
 - Semplice, 27
- Reti e matrici, 29
- Spazi
 - Lineare, 1
- Spazio
 - Hilbert, 3
 - normato, 2
- Spettro, 5
- Starting convenience, 52
- Subgraph centrality, 39
- Tre termini di ricorrenza, 15
- Unione, 28