



UNIVERSITÀ DEGLI STUDI DI CAGLIARI

FACOLTÀ DI SCIENZE

Corso di Laurea Magistrale in Matematica

Indirizzo Applicativo

Applicazione di algoritmi di Lanczos e formule
di quadratura all'analisi di reti complesse

Relatore:

Prof. Giuseppe Rodriguez

Candidato:

Marco Carta

Anno Accademico 2019/2020

19/05/2020

Indice

1	Polinomi Ortogonali	3
1.1	Relazione di ricorrenza	6
1.2	Proprietà degli zeri	9
1.3	Estensioni	10
2	Formule di Quadratura	11
2.1	Quadratura di Gauss	12
2.2	Quadratura di Gauss-Radau	14
2.3	Quadratura di Gauss-Lobatto	15
3	Spazi di Krylov	17
3.1	Algoritmo di Arnoldi	18
3.2	GMRES	22
3.3	Algoritmo di Lanczos	23
3.3.1	Legame con i polinomi ortogonali	24
3.4	Algoritmo di Lanczos non simmetrico	26
4	Grafi	32
4.1	Comunicabilità	32
4.2	Misure di centralità	34
5	Funzioni Matriciali	36
5.1	Utilizzo dell'algoritmo di Lanczos	36
5.2	Utilizzo dell'algoritmo di Lanczos non simmetrico	37
6	Risultati	42
	Bibliografia	48

Introduzione

Questa tesi nasce a seguito di un tirocinio formativo effettuato a fine 2019 presso l'Università di Cagliari sotto la guida del Prof. Rodriguez e del suo gruppo di lavoro, durante il quale il candidato ha intrapreso un percorso di ricerca incentrato sui polinomi ortogonali e i loro utilizzi in ambito numerico, tra le quali la stima di forme bilineari che si è scelto di approfondire al fine di analizzare reti complesse.

Il Capitolo 1 definisce i polinomi ortogonali ed elenca le loro caratteristiche fondamentali, quali la relazione di ricorrenza a tre termini e il loro legame con le matrici tridiagonali, da cui discendono le proprietà sugli zeri dei polinomi necessarie per costruire nel Capitolo 2 vari tipi di forme di quadratura per il calcolo di integrali definiti, imponendo la loro esattezza per l'integrazione di polinomi fino ad un certo grado fissato.

Nel Capitolo 3 vengono introdotti gli spazi di Krylov, strumenti storicamente utilizzati per la risoluzione di sistemi lineari tramite metodi di proiezione in spazi più piccoli rispetto a quello di lavoro. In tale contesto vengono descritti l'algoritmo di Arnoldi e i due di Lanczos, analizzando il loro legame con la teoria dei polinomi ortogonali.

Il concetto di rete viene introdotto nel Capitolo 4 dove si definisce la *subgraph centrality*, uno strumento utilizzato per l'analisi delle reti; questa quantità sarà calcolata nel Capitolo 5 tramite l'utilizzo di formule di quadratura derivanti dai metodi di Lanczos precedentemente descritti.

Infine nel Capitolo 6 si espongono i risultati prodotti da un'implementazione su MATLAB del procedimento discusso, analizzando reti di varia origine e conformazione in termini di prestazioni e affidabilità.

1 Polinomi Ortogonali

Per definire i polinomi ortogonali su un intervallo $[a, b]$ della retta reale occorre prima dare una definizione di ortogonalità. Una possibilità è ricorrere agli integrali di Riemann-Stieltjes per definire un prodotto scalare per funzioni di variabile reale, seguendo la trattazione proposta da Golub e Meurant in [1].

Definizione 1 (Integrale di Riemann-Stieltjes). *Data una funzione f continua e a valori reali, si denota il suo integrale di Riemann-Stieltjes su $[a, b]$ rispetto ad una funzione a valori reali α come*

$$\int_a^b f(\lambda) d\alpha(\lambda), \quad (1)$$

ed è definito come il limite della somma

$$\sum_{\lambda_i \in \pi} f(c_i)(\alpha(\lambda_{i+1}) - \alpha(\lambda_i)),$$

al tendere a zero del passo massimo della partizione π di $[a, b]$, con $c_i \in [\lambda_i, \lambda_{i+1}]$.

Nel caso particolare in cui $d\alpha(\lambda) = d\lambda$, l'integrale (1) è equivalente a quello di Riemann. Se α fosse di classe C^1 l'integrale sarebbe equivalente a

$$\int_a^b f(\lambda) \alpha'(\lambda) d\lambda,$$

ma questo non è in generale vero perché α potrebbe avere salti o derivata non continua: in tal caso l'integrale (1) non potrebbe essere scritto come integrale di Riemann. Mantenendo l'ipotesi $\alpha \in C^1$ scegliamo di rappresentarlo come

$$\int_a^b f(\lambda) \omega(\lambda) d\lambda, \quad (2)$$

dove ω viene detta funzione peso.

Il più semplice teorema di esistenza dell'integrale (1) richiede che f sia continua e α abbia “variazione limitata”, ovvero sia la differenza di due funzioni non decrescenti. In particolare il teorema è soddisfatto se α è non decrescente.

¹Una partizione di un intervallo $[a, b] \subset \mathbb{R}$ è un insieme π di $n+1$ punti c_i che lo dividono in sottointervalli, dunque si ha $\pi = \{c_i \in [a, b] \mid a = c_0 \leq \dots \leq c_n = b\}$.

Sia quindi α una funzione non decrescente nell'intervallo (a, b) , avente limiti finiti a $\pm\infty$ (nel caso l'intervallo non sia limitato) e con infiniti punti di incremento (si immagini α non continua, ma come funzione a gradini). Con le ipotesi appena definite diamo la seguente definizione:

Definizione 2 (Momenti di α). *I numeri*

$$\mu_i = \int_a^b \lambda^i d\alpha(\lambda)$$

sono detti **momenti** relativi alla misura α .

Per utilizzare l'integrale di Riemann-Stieltjes nella definizione dei polinomi ortogonali è necessario richiedere che tutti i momenti siano finiti. Con questa premessa si può definire un prodotto scalare attraverso un integrale della forma (1).

Definizione 3 (Prodotto Scalare). *Sia \mathcal{P} lo spazio dei polinomi reali. Si definisce il **prodotto scalare** relativo alla misura α dei polinomi $p, q \in \mathcal{P}$ come*

$$\langle p, q \rangle = \int_a^b p(\lambda)q(\lambda) d\alpha(\lambda). \quad (3)$$

Avere un prodotto scalare consente di definire uno spazio di Hilbert associato ad α

Definizione 4. *Data una misura α su $[a, b]$, si definisce lo spazio di Hilbert $L^2([a, b], \alpha)$ come*

$$L^2([a, b], \alpha) = \{ f \mid f \text{ misurabile, } \|f\|_\alpha < \infty \},$$

dove $\|f\|_\alpha$ è la norma indotta dal prodotto scalare (3), esteso alle funzioni

$$\|f\|_\alpha = \langle f, f \rangle^{1/2} = \left(\int_a^b f^2(\lambda) d\alpha(\lambda) \right)^{1/2}.$$

Si può considerare un'approssimazione discreta del prodotto scalare (3) attraverso la formula

$$\langle p, q \rangle = \sum_{j=1}^m p(t_j)q(t_j)\omega_j^2, \quad (4)$$

dove i valori t_j vengono denominati nodi e gli ω_j^2 pesi. La sommatoria può anche essere interpretata come un integrale di Stieltjes di una misura *discreta* a gradini con salti in

corrispondenza dei nodi t_j :

$$\alpha(\lambda) = \begin{cases} 0 & \lambda < t_1, \\ \sum_{j=1}^i \omega_j^2 & t_i \leq \lambda \leq t_{i+1} \quad \text{con } i = 1, \dots, m-1, \\ \sum_{j=1}^m \omega_j^2 & t_m \leq \lambda. \end{cases} \quad (5)$$

È ora possibile dare le seguenti definizioni:

Definizione 5 (Polinomi Ortogonali). *Due polinomi p e q sono detti **ortogonali** rispetto alle formule (3) o (4) se*

$$\langle p, q \rangle = 0.$$

Definizione 6 (Polinomi Ortonormali). *I polinomi di un insieme $\mathcal{A} \subset \mathcal{P}$ sono detti **ortonormali** se sono mutualmente ortogonali e si ha*

$$\|p\|_\alpha = 1, \quad \forall p \in \mathcal{A},$$

dove con $\|p\|_\alpha$ si indica la norma indotta dal prodotto scalare.

Definizione 7 (Polinomi Ortogonali Monici). *I polinomi di un insieme $\mathcal{A} \subset \mathcal{P}$ sono detti **polinomi ortogonali monici** se sono ortonormali, monici² e*

$$\|p\|_\alpha > 0, \quad \forall p \in \mathcal{A}.$$

Una condizione necessaria e sufficiente affinché il prodotto scalare (3) sia definito positivo richiede che siano positivi i determinanti delle *matrici di Hankel* dei momenti, definite per $k = 1, 2, \dots$, in questo modo:

$$H_k = \begin{bmatrix} \mu_0 & \mu_1 & \dots & \mu_{k-1} \\ \mu_1 & \mu_2 & \dots & \mu_k \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{k-1} & \mu_k & \dots & \mu_{2k-2} \end{bmatrix}.$$

Si può dunque enunciare il seguente Teorema.

Teorema 1. *Se il prodotto scalare (3) è definito positivo su \mathcal{P} , esiste un'unica sequenza infinita di polinomi ortogonali monici relativa alla misura α .*

²Un polinomio di variabile reale è detto **monico** se il coefficiente del termine di grado massimo è 1.

1.1 Relazione di ricorrenza

I polinomi ortogonali soddisfano una relazione di ricorrenza a tre termini che consente, se necessario, di avere un semplice metodo di calcolo per valutarli ed è alla base del legame con l'algoritmo di Lanczos che verrà discusso in seguito. La relazione si ricava per un qualsiasi prodotto scalare che soddisfi $\langle \lambda p, q \rangle = \langle p, \lambda q \rangle$, formula banalmente valida nel caso dell'integrale di Stieltjes (1). Vediamo nel dettaglio come si ricava la relazione nel caso dei polinomi ortogonali monici.

Teorema 2 (Relazione di ricorrenza). *I polinomi ortogonali monici possono essere ricavati dalla relazione di ricorrenza a tre termini*

$$\begin{cases} p_{k+1} = (\lambda - \alpha_{k+1})p_k - \beta_k p_{k-1}, & k = 0, 1, \dots \\ p_{-1}(\lambda) \equiv 0, & p_0(\lambda) \equiv 1 \end{cases} \quad (6)$$

con α_k e β_k successioni definite da

$$\alpha_{k+1} = \frac{\langle \lambda p_k, p_k \rangle}{\langle p_k, p_k \rangle}, \quad k = 0, 1, \dots$$

$$\beta_k = \frac{\langle p_k, p_k \rangle}{\langle p_{k-1}, p_{k-1} \rangle}, \quad k = 1, 2, \dots$$

Dimostrazione. Poiché i polinomi ortogonali sono linearmente indipendenti, ogni polinomio p di grado k può essere scritto come

$$p = \sum_{j=0}^k \delta_j p_j,$$

ed essendo monici il polinomio $p_{k+1} - \lambda p_k$ è di grado $\leq k$. Scriviamo la sua decomposizione nella forma

$$p_{k+1} - \lambda p_k = -\alpha_{k+1} p_k - \beta_k p_{k-1} + \sum_{j=0}^{k-2} \delta_j p_j. \quad (7)$$

È sufficiente provare che tutti i δ_j sono nulli. Moltiplicando scalarmente l'equazione per p_k e sfruttando l'ortogonalità si trova

$$\langle \lambda p_k, p_k \rangle = \alpha_{k+1} \langle p_k, p_k \rangle,$$

e di conseguenza, sfruttando il fatto che $\langle p_k, p_k \rangle$ è strettamente maggiore di 0 otteniamo

$$\alpha_{k+1} = \frac{\langle \lambda p_k, p_k \rangle}{\langle p_k, p_k \rangle}.$$

In maniera simile, moltiplicando la (7) per p_{k-1} si ottiene

$$\langle \lambda p_k, p_{k-1} \rangle = \beta_k \langle p_{k-1}, p_{k-1} \rangle,$$

e sfruttando il fatto che

$$\langle \lambda p_k, p_{k-1} \rangle = \langle p_k, \lambda p_{k-1} \rangle = \langle p_k, p_k + \sum_{j=0}^{k-1} \delta_j p_j \rangle = \langle p_k, p_k \rangle,$$

si ricava l'espressione per β_k

$$\beta_k = \frac{\langle p_k, p_k \rangle}{\langle p_{k-1}, p_{k-1} \rangle},$$

tenendo presente che non è necessario definire β_0 poichè $p_{-1}(\lambda) \equiv 0$. Infine moltiplicando ancora la (7) per p_j con $j < k - 1$ si ottiene

$$\langle \lambda p_k, p_j \rangle = \delta_j \langle p_j, p_j \rangle.$$

Ma

$$\langle \lambda p_k, p_j \rangle = \langle p_k, \lambda p_j \rangle = 0,$$

poichè λp_j è di grado minore di k . Di conseguenza si ha $\delta_j = 0$ per ogni j e si può definire la relazione di ricorrenza (6). □

Vale anche il teorema opposto, attribuito a Favard (si rimanda a [2]).

Teorema 3 (Teorema di Favard). *Se una sequenza di polinomi monici p_k , $k = 0, 1, \dots$, soddisfa (6) con coefficienti reali $\beta_k > 0$, allora esiste una misura positiva α tale che la sequenza p_k risulti ortogonale con il prodotto scalare definito dall'integrale di Riemann - Stieltjes rispetto alla misura α .*

Dimostrazione. Per una dimostrazione con carattere generale si rimanda a [3]. □

Vale una relazione di ricorrenza anche nel caso dei polinomi ortonormali, che si ricava in maniera analoga a quella generale.

Teorema 4. *I polinomi ortonormali posso essere ricavati dalla relazione di ricorrenza a tre termini*

$$\begin{cases} \sqrt{\beta_{k+1}}\pi_{k+1}(\lambda) = (\lambda - \alpha_{k+1})\pi_k(\lambda) - \sqrt{\beta_k}\pi_{k-1}(\lambda), & k = 0, 1, \dots \\ \pi_{-1}(\lambda) \equiv 0, \\ \pi_0(\lambda) \equiv \frac{1}{\sqrt{\beta_0}}, \\ \beta_0 = \int_a^b d\alpha, \end{cases} \quad (8)$$

con in particolare

$$\alpha_{k+1} = \langle \lambda \pi_k, \pi_k \rangle, \quad k = 0, 1, \dots,$$

e β_k calcolato in maniera tale da avere $\|\pi_k\|_\alpha = 1$.

Si può vedere facilmente che le due definizioni sono consistenti; se infatti si applica la (6) ad una sequenza di polinomi ortonormali si trovano esattamente i coefficienti definiti in (8). Viceversa per passare da polinomi ortogonali a ortonormali è sufficiente mettere sotto radici i coefficienti β_k . Per snellire la notazione a volte si pone $\sqrt{\beta_k} \stackrel{\text{not}}{=} \eta_k$. Nel seguito della trattazione si utilizzeranno principalmente i polinomi ortonormali; se questi esistono per ogni k , allora si può definire la matrice di Jacobi di ordine infinito

$$J_\infty = \begin{bmatrix} \alpha_1 & \sqrt{\beta_1} & & & \\ \sqrt{\beta_1} & \alpha_2 & \sqrt{\beta_2} & & \\ & \sqrt{\beta_2} & \alpha_3 & \sqrt{\beta_3} & \\ & & \ddots & \ddots & \ddots \end{bmatrix} \stackrel{\text{not}}{=} \begin{bmatrix} \alpha_1 & \eta_1 & & & \\ \eta_1 & \alpha_2 & \eta_2 & & \\ & \eta_2 & \alpha_3 & \eta_3 & \\ & & \ddots & \ddots & \ddots \end{bmatrix}, \quad (9)$$

che risulta di vitale importanza nella teoria dei polinomi ortogonali e per gli scopi di questa tesi consentirà di costruire un legame col calcolo delle funzioni matriciali. Per il minore principale di ordine k si utilizza la notazione J_k .

I polinomi ortogonali soddisfano un'importante proprietà di minimo. In [4] è dimostrato infatti il seguente teorema:

Teorema 5. *Si consideri lo spazio di Hilbert $L^2([a, b], \alpha)$ e si indichi con $\|\cdot\|_\alpha$ la sua norma. Se q_k è un polinomio monico di grado k , allora il minimo*

$$\min_{q_k} \{ \|q_k\|_\alpha \} = \min_{q_k} \left\{ \int_a^b q_k^2 d\alpha \right\}$$

è raggiunto dal polinomio ortogonale p_k relativo alla misura α .

Vale inoltre la formula di *Christoffel-Darboux*, utile per dimostrare altre proprietà dei polinomi ortonormali.

$$\sum_{i=0}^k \pi_i(\lambda)\pi_i(\mu) = \sqrt{\beta_{k+1}} \frac{\pi_{k+1}(\lambda)\pi_k(\mu) - \pi_k(\lambda)\pi_{k+1}(\mu)}{\lambda - \mu} \quad \text{se } \lambda \neq \mu, \quad (10)$$

$$\sum_{i=0}^k \pi_i(\lambda)^2 = \sqrt{\beta_{k+1}} \left[\pi'_{k+1}(\lambda)\pi_k(\lambda) - \pi'_k(\lambda)\pi_{k+1}(\lambda) \right] \quad \text{nel caso } \lambda = \mu. \quad (11)$$

1.2 Proprietà degli zeri

Consideriamo i primi k polinomi ortonormali e fissiamo λ per costruire il vettore

$$P_k(\lambda) = (p_0(\lambda), p_1(\lambda), \dots, p_{k-1}(\lambda))^T.$$

Utilizziamo il vettore per scrivere la relazione di ricorrenza in forma matriciale:

$$\lambda P_k(\lambda) = J_k P_k + \sqrt{\beta_k} p_k(\lambda) e^k, \quad (12)$$

dove J_k è la matrice di Jacobi e e^k è un vettore contenente la k -esima colonna della matrice identità di ordine k . Questa scrittura conduce ad un risultato fondamentale.

Teorema 6. *Gli zeri $\theta_j^{(k)}$ del polinomio ortonormale p_k sono gli autovalori della matrice di Jacobi J_k .*

Dimostrazione. Se θ è uno zero di p_k , l'equazione (12) diventa

$$\theta P_k(\theta) = J_k P_k(\theta),$$

e per definizione di autovalore si ha la tesi. □

Si può ora enunciare il risultato fondamentale riguardante gli zeri dei polinomi ortogonali.

Teorema 7. *Gli zeri dei polinomi ortogonali su $[a, b]$ associati alla misura α sono reali, distinti e appartenenti all'interno dell'intervallo $[a, b]$.*

Dimostrazione. Poiché la matrice di Jacobi è simmetrica, tridiagonale e irriducibile, i suoi autovalori sono reali e distinti e questa proprietà si riflette sugli zeri dei polinomi. Per la dimostrazione della seconda parte del problema si rimanda a [4]. □

1.3 Estensioni

Si può costruire una teoria sui polinomi ortogonali anche nel caso in cui α non sia una misura positiva, in quanto definita a partire da una funzione peso che cambia segno in $[a, b]$: questa teoria è discussa in [5] e sarà brevemente introdotta nel seguito di questa tesi allo scopo di descrivere l'Algoritmo di Lanczos non simmetrico e il suo utilizzo pratico ai fini di questo lavoro. Una ulteriore generalizzazione consiste nel costruire dei polinomi dove i coefficienti non sono numeri reali ma matrici. Tali polinomi sono detti *matriciali* e sono discussi brevemente in [1], dove sono presenti le opportune citazioni per approfondire l'argomento. Il loro utilizzo consente di ottimizzare l'utilizzo dell'Algoritmo di Lanczos non simmetrico proposto in questa tesi e di non incorrere nelle sue limitazioni, come descritto in [6] e [7]. Per alcune scelte *classiche* di $[a, b]$ e della misura α^3 i coefficienti della relazione di ricorrenza possono essere calcolati per mezzo di formule chiuse e i corrispondenti polinomi ortogonali monici sono di conseguenza facilmente calcolabili.

$\omega(\lambda)$	Supporto	Nome	α_k	β_0	$\beta_k, k \geq 1$
1	$[-1, 1]$	Legendre	0	2	$\frac{1}{4-k^2}$
1	$[0, 1]$	Legendre Shiftati	$\frac{1}{2}$	1	$\frac{1}{4(4-k^2)}$
$(1-t^2)^{-\frac{1}{2}}$	$[-1, 1]$	Chebyshev I tipo	0	π	$\frac{1}{2} (k=1) \frac{1}{4} (k \neq 1)$
$(1-t^2)^{\frac{1}{2}}$	$[-1, 1]$	Chebyshev II tipo	0	$\frac{\pi}{2}$	$\frac{1}{4}$
e^{-t}	$[0, \infty]$	Laguerre	$2k+1$	1	k^2
e^{-t^2}	$[-\infty, \infty]$	Hermite	0	$\sqrt{\pi}$	$\frac{k}{2}$
$(1-t)^\alpha(1+t)^\beta$	$[-1, 1]$	Jacobi	α_k^J	β_0^J	β_k^J
Per i polinomi di Jacobi $\alpha, \beta > -1$. I coefficienti della relazione valgono:					
$\alpha_k^J = \frac{\beta^2 - \alpha^2}{(2k + \alpha + \beta)(2k + \alpha + \beta + 2)}, \quad \beta_0^J = \frac{2^{\alpha+\beta+1}\Gamma(\alpha+1)\Gamma(\beta+1)}{\Gamma(\alpha+\beta+2)}, \quad \beta_k^J = \frac{4k(k+\alpha)(k+\beta)(k+\alpha+\beta)}{(2k+\alpha+\beta)^2(2k+\alpha+\beta+1)(2k+\alpha+\beta-1)}$					

Tabella 1: Polinomi Ortogonali principali relativi alla notazione $d\alpha(\lambda) = \omega(\lambda)d\lambda$.

³Negli esempi riportati in realtà non si sceglie α bensì una funzione peso misurabile ω che definisce un integrale di Riemann-Stieltjes della forma (2). La misura α è dunque una primitiva della funzione peso ω .

2 Formule di Quadratura

Data una misura α sull'intervallo $[a, b]$ e una funzione f tali che esistano l'integrale di Stieltjes e tutti i momenti, una formula di quadratura è un'espressione dell'integrale stesso, che assume la forma

$$\int_a^b f(\lambda) d\alpha = \sum_{j=1}^N \omega_j f(t_j) + R[f]. \quad (13)$$

La sommatoria è utilizzata per approssimare l'integrale mentre $R[f]$ è il relativo residuo. I t_j sono detti *odi* e gli ω_j pesi. La formula è detta di grado d se $R[p] = 0$ per ogni polinomio p di grado $\leq d$ e $R[q] \neq 0$ per almeno un polinomio q di grado $d + 1$. Si possono ottenere formule di quadratura di grado $N - 1$, approssimando f con un polinomio interpolante. Ad esempio nel caso del polinomio interpolante di Lagrange si ha

$$f(\lambda) \approx \sum_{i=1}^N f(t_i) l_i(\lambda), \quad \text{con}$$

$$l_i(\lambda) = \prod_{\substack{j=1 \\ j \neq i}}^N \frac{\lambda - t_j}{t_i - t_j}.$$

Si procede quindi sostituendo l'approssimazione polinomiale nell'integrale per calcolare i pesi:

$$\omega_i = \int_a^b l_i(\lambda) d\alpha(\lambda).$$

Per ottenere formule di quadratura di grado superiore si utilizzano i polinomi ortogonali; il primo mezzo utile per comprendere questo legame è il seguente teorema, dimostrato da Gautschi in [8].

Teorema 8. *Sia $k \leq N$. La formula (13) ha grado $d = N - 1 + k$ se e soltanto se è interpolatoria⁴ e*

$$\int_a^b \prod_{j=1}^N (\lambda - t_j) p(x) d\alpha = 0,$$

per ogni polinomio p di grado minore o uguale a $k - 1$.

⁴Una formula di quadratura è detta *interpolatoria* se può essere ottenuta integrando un'approssimazione della funzione f data da un polinomio interpolante, come ad esempio quello di Lagrange.

Il caso $k = N$ risulta limite per il teorema perché α è positiva: infatti se si avesse $k = N + 1$ il teorema implicherebbe l'ortogonalità tra il polinomio

$$\prod_{j=1}^N (\lambda - t_j)$$

e sé stesso. La quadratura ottimale ha dunque grado $2N - 1$, è detta *Quadratura di Gauss* ed è apparsa per la prima volta a inizio ottocento in [9]; per poterla utilizzare è necessario un metodo per calcolare i nodi e i pesi a partire dall'espressione della formula stessa. Affrontiamo questo problema esprimendo in generale una formula di quadratura come un'espressione dell'integrale di Stieltjes con questa forma:

$$\int_a^b f(\lambda) d\alpha = \sum_{j=1}^N \omega_j f(t_j) + \sum_{k=1}^M v_k f(z_k) + R[f]. \quad (14)$$

dove i nodi z_k sono fissati a priori. È il caso delle formule di Gauss-Radau (in tal caso si fissa come nodo un solo estremo dell'intervallo $[a, b]$) e di Gauss-Lobatto (si fissano entrambi gli estremi). Per stimare l'accuratezza di una formula di quadratura ci si avvale della seguente espressione del residuo, valida per una misura α positiva non decrescente e una funzione $f \in C^{(2N+M)}$ (dettagli in [10]):

$$R[f] = \frac{f^{(2N+M)}(\eta)}{(2N+M)!} \int_a^b \prod_{k=1}^M (\lambda - z_k) \left[\prod_{j=1}^N (\lambda - t_j) \right]^2 d\alpha, \quad a < \eta < b. \quad (15)$$

2.1 Quadratura di Gauss

Soffermiamoci ora sul caso in cui $M = 0$, e dunque sulla *Quadratura di Gauss*. I pesi e i nodi potrebbero essere calcolati a partire dai momenti e risolvendo un sistema non lineare; ovviamente questa metodologia non è molto pratica e si ricorre al Teorema 1 e alle sue generalizzazioni nel caso ortonormale. La teoria dei polinomi ortogonali consente infatti di calcolare nodi e pesi in maniera agevole: utilizzando il teorema di esistenza e gli altri strumenti discussi in precedenza, quali la relazione di ricorrenza (8), la matrice di Jacobi e la caratterizzazione dei nodi si arriva al seguente teorema.

Teorema 9. *Gli autovalori della matrice di Jacobi J_N , e dunque gli zeri dei polinomi ortonormali associati ad α , sono i nodi della formula di quadratura di Gauss; i pesi della formula sono dati dai quadrati delle prime componenti dei corrispondenti autovettori normalizzati.*

Dimostrazione. Dal Teorema 8, il polinomio monico $\prod_{j=1}^N (\lambda - t_j)$ è ortogonale a tutti i polinomi di grado $k - 1 = N - 1$. Di conseguenza, è uguale per definizione al polinomio ortogonale di grado k che si costruisce con la formula di ricorrenza a partire dalla misura α . Avendolo espresso come produttoria è evidente che i suoi zeri sono esattamente i nodi della formula di quadratura di Gauss t_j , ma dalla teoria sui polinomi ortogonali è noto che questi sono anche uguali agli autovalori della matrice di Jacobi J_N e la prima parte del teorema è dimostrata. Consideriamo quindi un autovettore non normalizzato della matrice di Jacobi, corrispondente all'autovalore t_j , e denotiamolo con $P(t_j)$. Se q è un autovettore unitario, si ha che $P(t_j) = cq$. Dalla formula di *Christoffel-Darboux* (11) si può ricavare (dettagli in [11] e [12])

$$\omega_j P(t_j)^T P(t_j) = 1, \quad j = 1, \dots, N.$$

ma poichè

$$\omega_j P(t_j)^T P(t_j) = \omega_j c^2 \|q\|^2 = \omega_j c^2,$$

ricaviamo

$$\omega_j c^2 = 1 \implies \omega_j = \frac{1}{c^2}.$$

Per trovare c , si può considerare una qualsiasi componente dell'autovettore q che sia diversa da zero, ad esempio la prima. Da $P(t_j) = cq$ si ha

$$c = \frac{\pi_0(t_j)}{q_1} = \frac{1}{q_1},$$

in quanto $\pi_0 \equiv 1$. Sostituendo nell'espressione di ω_j otteniamo

$$\omega_j = q_1^2.$$

Si noti che se l'integrale della misura non fosse identicamente uguale a 1, non si avrebbe $\pi_0 \equiv 1$ e quindi

$$\omega_j = q_1^2 \mu_0 = q_1^2 \int_a^b d\alpha.$$

□

Il teorema consente quindi di calcolare nodi e pesi della formula di quadratura di Gauss a partire dalla matrice di Jacobi. La struttura della matrice rende il calcolo degli autovalori

molto efficiente, con l'utilizzo di metodi di tipo QR specifici per matrici tridiagonali. I pesi possono essere calcolati senza passare per il calcolo completo degli autovettori, con un metodo apposito proposto da Golub e Welsch in [12].

Indicando i pesi con ω_j^G e i nodi con t_j^G le formule (14) e (15) diventano:

$$I[f] = \int_a^b f(\lambda) d\alpha = \sum_{j=1}^N \omega_j^G f(t_j^G) + R_G[f], \quad (16)$$

$$R_G[f] = \frac{f^{(2N)}(\eta)}{(2N)!} \int_a^b \left[\prod_{j=1}^N (\lambda - t_j^G) \right]^2 d\alpha. \quad (17)$$

Vale inoltre il seguente Teorema che consente di dare una stima in valore assoluto dell'errore.

Teorema 10. *Supponiamo di avere $\forall n, \xi, f^{(2n)}(\xi) > 0$ con $a < \xi < b$ e definiamo*

$$L_G[f] = \sum_{j=1}^N \omega_j^G f(t_j^G).$$

Allora tale formula (di Gauss) è esatta per tutti i polinomi di grado minore o uguale a $2N - 1$ e si ha

$$L_G[f] \leq I[f].$$

Inoltre $\forall N, \exists \eta \in [a, b]$ tale che

$$I[f] - L_G[f] = \frac{f^{(2N)}(\eta)}{(2N)!} \prod_{j=1}^{N-1} \beta_j^2.$$

Dal teorema si nota quindi che per avere una stima dell'errore è sufficiente avere una stima della derivata $2N$ -esima di f . Si possono ottenere risultati simili ponendo $M = 1, 2$ nell'equazione (14). È il caso delle formule di *Gauss-Radau* e *Gauss-Lobatto*. Si rimanda a [1] e [13] per un'analisi più approfondita.

2.2 Quadratura di Gauss-Radau

La formula di Gauss-Radau è ottenuta fissando come nodo uno degli estremi dell'intervallo $[a, b]$, avendo quindi $M = 1$ in (14). È costruita estendendo la matrice di Jacobi, imponendo quindi che abbia un nuovo autovalore prefissato; per farlo è necessario risolvere un sistema

tridiagonale, quindi non è un passaggio troppo costoso dal punto di vista computazionale. Il Teorema 10 viene generalizzato opportunamente:

Teorema 11. *Supponiamo di avere $\forall n, \xi, f^{(2n+1)}(\xi) < 0$ con $a < \xi < b$ e utilizzando i nodi e i pesi ottenuti estendendo la matrice J_N definiamo*

$$U_{GR}[f] = \sum_{j=1}^N \omega_j^a f(t_j^a) + v_1^a f(a),$$

$$L_{GR}[f] = \sum_{j=1}^N \omega_j^b f(t_j^b) + v_1^b f(b).$$

Allora tale formula (di Gauss-Radau) è esatta per tutti i polinomi di grado minore o uguale a $2N$ e si ha

$$L_{GR}[f] \leq I[f] \leq U_{GR}[f].$$

Inoltre $\forall N, \exists \eta_U, \eta_L \in [a, b]$ tali che

$$I[f] - U_{GR}[f] = \frac{f^{(2N+1)}(\eta_U)}{(2N+1)!} \int_a^b (\lambda - a) \left[\prod_{j=1}^N (\lambda - t_j^a) \right]^2 d\alpha,$$

$$I[f] - L_{GR}[f] = \frac{f^{(2N+1)}(\eta_L)}{(2N+1)!} \int_a^b (\lambda - b) \left[\prod_{j=1}^N (\lambda - t_j^b) \right]^2 d\alpha.$$

Se l'ipotesi del Teorema fosse del tipo $f^{(2n+1)}(\xi) > 0$ le stime sarebbero invertite.

2.3 Quadratura di Gauss-Lobatto

La formula di Gauss-Lobatto si ricava in modo simile, con l'idea di fissare come nodi entrambi gli estremi dell'intervallo $[a, b]$ in (14). In questo caso si deve estendere la matrice di Jacobi imponendo due nuovi autovalori, dopo aver risolto due sistemi tridiagonali. Il Teorema 10 si trasforma ancora:

Teorema 12. *Supponiamo di avere $\forall n, \xi, f^{(2n)}(\xi) < 0$ con $a < \xi < b$ e utilizzando i nodi e i pesi ottenuti estendendo la matrice J_N definiamo*

$$U_{GL}[f] = \sum_{j=1}^N \omega_j^{GL} f(t_j^{GL}) + v_1^{GL} f(a) + v_2^{GL} f(b).$$

Allora tale formula (di Gauss-Lobatto) è esatta per tutti i polinomi di grado minore o uguale a $2N + 1$ e si ha

$$I[f] \leq U_{GL}[f].$$

Inoltre $\forall N, \exists \eta \in [a, b]$ tale che

$$I[f] - U_{GL}[f] = \frac{f^{(2N+2)}(\eta)}{(2N+2)!} \int_a^b (\lambda - a)(\lambda - b) \left[\prod_{j=1}^N (\lambda - t_j^{GL}) \right]^2 d\alpha.$$

Confrontando l'enunciato del Teorema nella sua versione originale per la formula di Gauss con le sue varianti, si nota che nella formula di Gauss-Radau si guadagna un ordine di accuratezza (essendo l'errore dipendente dalla derivata $(N+1)$ -esima) e si possono avere una stima dal basso e una dall'alto del risultato esatto dell'integrale, a seconda del nodo scelto. Nel caso della formula di Gauss-Lobatto invece si guadagnano due ordini di accuratezza e l'errore ha lo stesso segno di quello della formula di Gauss.

3 Spazi di Krylov

Data una matrice quadrata A di ordine n e un vettore $v \in \mathbb{R}^n$, uno spazio di Krylov di ordine m è definito come

$$K_m(A, v) = \text{span} \{ v, Av, \dots, A^{m-1}v \}, \quad (18)$$

dunque un suo elemento può essere scritto nella forma $x = q_{m-1}(A)v$, dove q_{m-1} è un polinomio di grado $m - 1$.

Se la matrice A è diagonalizzabile la successione $A^k v$ è convergente per $k \rightarrow \infty$, in quanto dal metodo delle potenze è noto che converge a un autovettore (relativo all'autovalore di modulo massimo): questa osservazione ha conseguenze, in quanto i vettori della successione tendono a diventare paralleli tra loro. Ciò comporta che qualora tali vettori fossero linearmente indipendenti in aritmetica esatta, potrebbero risultare dipendenti in fase di implementazione a causa degli errori di arrotondamento prodotti dall'aritmetica di macchina. La successione utilizzata nella definizione (18) non si presta quindi, per sua natura, ad essere utilizzata nelle implementazioni. Può anche capitare che, per un certo M , si abbia $A^M v = 0$ in aritmetica esatta (M prende il nome di *grado* di v rispetto ad A) e di conseguenza la successione non possa proseguire. Questa casistica non è un problema puramente numerico ed è favorevole o sfavorevole a seconda dell'applicazione per la quale si utilizzano questi spazi. Gli spazi di Krylov possono essere utilizzati nella risoluzione tramite *metodi di proiezione* (si rimanda a [14] per dettagli sulla teoria generale) del sistema lineare

$$Ax = b.$$

In generale un metodo di proiezione ricerca, a partire da una soluzione iniziale x_0 , una soluzione approssimata x_m in un sottospazio affine di dimensione m $\{x_0 + K_m\}$ imponendo la condizione di ortogonalità di Petrov-Galerkin

$$b - Ax_m \perp L_m, \quad (19)$$

dove L_m è un altro sottospazio di dimensione m . Questa condizione esprime la necessità di avere m condizioni di ortogonalità in maniera da scegliere una soluzione $x_m \in \{x_0 + K_m\}$.

In un metodo di proiezione che utilizza spazi di Krylov si ha di solito

$$\begin{cases} K_m = K_m(A, r_0) = \text{span} \{ r_0, Ar_0, \dots, A^{m-1}r_0 \}, \\ r_0 = b - Ax_0. \end{cases}$$

Ciò che differenzia un metodo dall'altro dal punto di vista teorico è dunque la scelta del sottospazio L_m , della soluzione iniziale x_0 ed eventualmente l'utilizzo di preconditionatori. Ricordando la definizione di spazio di Krylov possiamo esprimere la soluzione approssimante x_m come

$$A^{-1}b \approx x_m = x_0 + q_{m-1}(A)r_0,$$

che diventa, nel caso più semplice in cui $x_0 = 0$,

$$A^{-1}b \approx x_m = q_{m-1}(A)b.$$

Per effettuare a livello pratico la proiezione su K_m si possono seguire diversi algoritmi, che cercano di costruire una base di utilizzo più agevole rispetto a quella data dalla definizione (18).

3.1 Algoritmo di Arnoldi

Un primo approccio è dato dall'Algoritmo di Arnoldi [15], che non richiede ipotesi aggiuntive sulla matrice A per costruire una base ortonormale di $K_m(A, v)$. L'Algoritmo si inizializza normalizzando il vettore v , ponendo $v^1 = \frac{v}{\|v\|}$. Ad ogni passo k si genera v^{k+1} basandosi sull'equazione matriciale

$$AV_k = V_k H_k + h_{k+1,k} v^{k+1} (e^k)^T, \quad (20)$$

dove e^k indica la k -esima colonna della matrice identità di ordine n , V_k è una matrice avente per colonne i vettori della base ortonormale di $K_k(A, v)$ che si vuole costruire

$$V_k = \begin{bmatrix} v^1 & \dots & v^k \end{bmatrix},$$

e H_k è una matrice quadrata avente forma di Hessenberg

$$H_k = \begin{bmatrix} h_{11} & h_{12} & \cdots & \cdots & h_{1k} \\ h_{21} & h_{22} & \cdots & \cdots & h_{2k} \\ & h_{32} & \ddots & & h_{3k} \\ & & \ddots & \ddots & \vdots \\ & & & h_{k,k-1} & h_{kk} \end{bmatrix}.$$

Supponendo per il momento che i vettori che compongono V_k siano effettivamente una base, possiamo notare come l'equazione (20) descriva il comportamento della matrice A (o per meglio dire dell'operatore A) sullo spazio di Krylov, e dunque possa essere associata geometricamente ad un metodo di proiezione.

Esplicitando i conti nell'equazione (20) si ottiene

$$Av^k = \sum_{i=1}^k h_{ik} v^i + h_{k+1,k} v^{k+1}, \quad (21)$$

che una volta rielaborata introducendo la notazione

$$\tilde{v}^k = h_{k+1,k} v^{k+1} = Av^k - \sum_{i=1}^k h_{ik} v^i,$$

consente di esplicitare v^{k+1} ottenendo

$$v^{k+1} = \frac{\tilde{v}^k}{h_{k+1,k}} \implies h_{k+1,k} = \|\tilde{v}^k\|,$$

in quanto v^{k+1} deve essere di norma unitaria.

Per calcolare \tilde{v}^k , avendo già a disposizione i vari v^i , è necessario conoscere i coefficienti h_{ik} . Moltiplicando scalarmente la formula (21) per v^j e sfruttando l'ortonormalità si ottiene

$$h_{jk} = \langle Av^k, v^j \rangle, \quad j = 1, \dots, k.$$

Le formule ricavate definiscono quindi da un punto di vista operativo un'iterazione dell'Algoritmo di Arnoldi. I vettori così costruiti formano una base ortonormale per lo spazio di Krylov $K_m(A, v)$.

Infatti si dimostra per induzione che ogni vettore v^j è della forma $q_{j-1}(A)v^1$: ciò è banalmente vero nel caso $j = 1$, in quanto $v^1 = q_0(A)v^1$ con $q_0 \equiv \frac{1}{\|v\|}$. Supponendo che

questo sia vero fino a j , consideriamo v^{j+1} . Dalla formula (21) si ha:

$$h_{j+1,j}v^{j+1} = Av^j - \sum_{i=1}^j h_{ij}v^i = Aq_{j-1}(A)v^1 - \sum_{i=1}^j h_{ij}q_{i-1}(A)v^1 = q_j(A)v^1,$$

e quindi i vettori costruiti generano lo spazio di Krylov, che per definizione è generato da vettori aventi quella forma.

L'algoritmo di Arnoldi relativo a $K_m(A, v)$ si può dunque schematizzare, in aritmetica esatta, come indicato nella Figura 1.

Algoritmo 1: Algoritmo di Arnoldi

```

 $v^1 = v/\|v\|;$ 
for  $j = 1, \dots, m$  do
  for  $i = 1, \dots, j$  do
     $h_{ij} = \langle Av^j, v^i \rangle;$ 
  end
   $\tilde{v}^j = Av^j - \sum_{i=1}^j h_{ij}v^i;$ 
   $h_{j+1,j} = \|\tilde{v}^j\|;$ 
  if  $h_{j+1,j} = 0$  then
    stop (breakdown)
  end
  else
     $v^{j+1} = \tilde{v}^j/h_{j+1,j};$ 
  end
end

```

Figura 1: Algoritmo di Arnoldi in aritmetica esatta, a partire da un vettore v .

Si può osservare l'equivalenza tra l'Algoritmo di Arnoldi e il processo di ortogonalizzazione di Gram-Schmidt applicato alla base di K_m fornita dalla definizione (18). Da un punto di vista implementativo esistono diverse versioni dell'Algoritmo di Arnoldi, equivalenti a diverse versioni dell'ortogonalizzazione di Gram-Schmidt, descritte in [14].

Se nell'esecuzione dell'Algoritmo avviene un *breakdown*, si è raggiunta la massima dimensione possibile di K_m e l'algoritmo non può continuare in quanto lo spazio di Krylov è

invariante rispetto all'operatore A . Se si usa l'Algoritmo di Arnoldi per la risoluzione di un sistema lineare tramite un metodo di proiezione il breakdown è un evento positivo, in quanto significa che il sottospazio $K_m \subset \mathbb{R}^n$ contiene la soluzione esatta del sistema. Si può quindi definire facilmente un metodo di proiezione ortogonale, con $L_m = K_m = x_0 + K_m(A, r_0)$. Ponendo infatti $\beta = \|r_0\|$ e

$$v^1 = \frac{r_0}{\beta}$$

come vettore iniziale, si ha dopo m iterazioni

$$\begin{cases} V_m^T A V_m = H_m, \\ V_m^T r_0 = V_m^T (\beta v^1) = \beta e_1. \end{cases}$$

Di conseguenza si può scegliere una soluzione approssimante nel sottospazio di dimensione m , ponendo

$$\begin{cases} x_m = x_0 + V_m y_m, \\ y_m = H_m^{-1}(\beta e_1). \end{cases}$$

Questa scelta rispetta la condizione di ortogonalità di Petrov-Galerkin (19) in quanto

$$\begin{aligned} b - Ax_m &= b - A(x_0 + V_m y_m), \\ &= r_0 - AV_m y_m, \\ &= \underbrace{\beta v_1 - V_m H_m y_m}_{=0} - h_{m+1,m} e_m^T y_m v_{m+1}, \\ &= \gamma v_{m+1}. \end{aligned}$$

e v_{m+1} è ortogonale a K_m per definizione.

L'Algoritmo di Arnoldi può inoltre essere utilizzato per stimare gli autovalori estremali della matrice A , in quanto quelli di H_m ne costituiscono una stima. In questa applicazione un breakdown non è una buona notizia, in quanto non consente la prosecuzione dell'algoritmo per migliorare la stima degli autovalori. Si può aggirare il problema facendo ripartire l'Algoritmo con un diverso vettore iniziale.

3.2 GMRES

Un metodo GMRES (*Generalized Minimum Residual*) è un metodo di proiezione per sistemi lineari basato sullo scegliere

$$\begin{cases} K_m = K_m(A, r_0), \\ L_m = AK_m, \end{cases}$$

per ottenere una soluzione che minimizzi la norma del residuo in $x_0 + K_m$. La sua versione più semplice si basa sull'Algoritmo di Arnoldi, scrivendo l'equazione (20) nella forma

$$AV_m = V_{m+1}\tilde{H}_m, \quad (22)$$

dove la matrice \tilde{H}_m è in forma di Hessenberg ma non è più quadrata, come nei precedenti metodi di proiezione illustrati. Dall'equazione (22) si ricava

$$\begin{aligned} b - Ax_m &= b - A(x_0 + V_my_m), \\ &= r_0 - AV_my_m, \\ &= \beta v_1 - V_{m+1}\tilde{H}_my_m, \\ &= V_{m+1}(\beta e_1 - \tilde{H}_my_m). \end{aligned}$$

Poichè le colonne di V_{m+1} sono ortonormali e $b \in \mathcal{R}(V_{m+1})$ ha

$$\|b - Ax_m\|_2 = \|\beta e_1 - \tilde{H}_my_m\|_2.$$

Di conseguenza il metodo GMRES è basato sulle equazioni

$$\begin{cases} x_m = x_0 + V_my_m, \\ y_m = \underset{y \in \mathbb{R}^m}{\operatorname{argmin}} \|\beta e_1 - \tilde{H}_my_m\|_2. \end{cases}$$

Dal momento che \tilde{H}_m è sovradeterminata si deve quindi risolvere un problema ai minimi quadrati invece del solito sistema lineare ben posto. Questo procedimento può essere modificato utilizzando le versioni dell'Algoritmo di Arnoldi esistenti in letteratura, come descritto in [14].

3.3 Algoritmo di Lanczos

L'Algoritmo di Lanczos, apparso per la prima volta in [16] antecedentemente a quello di Arnoldi [15], può essere ricavato applicando l'algoritmo appena descritto ad una matrice simmetrica.

Scegliamo però di trattarlo individualmente, ripercorrendo il ragionamento precedente. L'obiettivo è dunque costruire una base ortonormale del sottospazio di Krylov $K_m(A, v)$, con A matrice simmetrica. L'algoritmo si inizializza ponendo $v^1 = \frac{v}{\|v\|}$, e ad ogni passo k genera v^{k+1} basandosi su un'equazione matriciale simile alla (20):

$$AV_k = V_k J_k + \eta_{k+1} v^{k+1} (e^k)^T, \quad (23)$$

ma stavolta al posto di H_k si ha J_k , avente struttura tridiagonale simmetrica

$$J_k = \begin{bmatrix} \alpha_1 & \eta_2 & & & \\ \eta_2 & \alpha_2 & \eta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \eta_{k-1} & \alpha_{k-1} & \eta_k \\ & & & \eta_k & \alpha_k \end{bmatrix}, \quad (24)$$

che descriviamo utilizzando α_k, η_k in linea con la notazione introdotta nel Capitolo 1. A partire dall'equazione cerchiamo dunque di ricavare v^{k+1} . Esplicitando il calcolo si ha

$$Av^k = \eta_k v^{k-1} + \alpha_k v^k + \eta_{k+1} v^{k+1}, \quad (25)$$

che anche in questo caso possiamo rielaborare introducendo la notazione

$$\tilde{v}^k = \eta_{k+1} v^{k+1} = Av^k - \eta_k v^{k-1} - \alpha_k v^k,$$

per esplicitare v^{k+1} nella forma

$$v^{k+1} = \frac{\tilde{v}^k}{\eta_{k+1}} \implies \eta_{k+1} = \|\tilde{v}^k\|,$$

in quanto si richiede che v^{k+1} abbia norma unitaria. Per ricavare gli α_k è sufficiente moltiplicare scalarmente la (25) per v^k e sfruttare l'ortonormalità, ottenendo

$$\alpha_k = \langle Av^k, v^k \rangle.$$

Si può dimostrare, con lo stesso procedimento utilizzato nell'algoritmo di Arnoldi, che i vettori così costruiti generano dopo m iterazioni lo spazio di Krylov $K_m(A, v)$.

Algoritmo 2: Algoritmo di Lanczos

$$v^1 = v/\|v\|, \quad \eta_1 \equiv 0, \quad v^0 \equiv 0;$$

for $j = 1, \dots, m$ **do**

$$\alpha_j = \langle Av^j, v^j \rangle;$$

$$\tilde{v}^j = Av^j - \eta_j v^{j-1} - \alpha_j v^j;$$

$$\eta_{j+1} = \|\tilde{v}^j\|;$$

if $\eta_{j+1} = 0$ **then**

 | **stop** (breakdown)

end

else

$$\quad | \quad v^{j+1} = \tilde{v}^j / \eta_{j+1};$$

end

end

Figura 2: Algoritmo di Lanczos in aritmetica esatta, a partire da un vettore v .

3.3.1 Legame con i polinomi ortogonali

Il passo iterativo dell'algoritmo può essere espresso tramite la relazione di ricorrenza a tre termini

$$\eta_{j+1}v^{j+1} = Av^j - \alpha_j v^j - \eta_j v^{j-1}, \quad (26)$$

che è la stessa definita nel Capitolo 1 per i polinomi ortonormali, con la notazione già menzionata $\sqrt{\beta_k} \stackrel{\text{not}}{=} \eta_k$. Ricordando che si ha $v^i = q_{i-1}(A)v^1$, possiamo definire un prodotto scalare in \mathbb{P}_{m-1} ponendo

$$\langle p, q \rangle_v = \langle p(A)v, q(A)v \rangle. \quad (27)$$

Se lo spazio K_m ha dimensione massima questo prodotto scalare è non degenere e l'ortogonalità dei polinomi e quella dei v^i sono concetti equivalenti. L'Algoritmo di Lanczos è di fatto equivalente all'algoritmo di Stieltjes descritto da Gautschi in [8], che genera polinomi ortonormali a partire dal prodotto scalare (27).

Si arriva ai seguenti teoremi, dimostrati da Golub in [1]:

Teorema 13. Sia $\chi_k(\lambda)$ il determinante di $J_k - \lambda I$; allora

$$\begin{cases} v^k = p_k(A)v^1, \\ p_k(\lambda) = (-1)^{k-1} \frac{\chi_{k-1}(\lambda)}{\eta_1 \cdots \eta_{k-1}}, \\ k > 1, \quad p_1 \equiv 1. \end{cases}$$

I polinomi p_k hanno grado $k - 1$ e sono chiamati polinomi di Lanczos normalizzati.

Tali polinomi soddisfano dunque una relazione di ricorrenza,

$$\eta_k p_{k+1}(\lambda) = (\lambda - \alpha_k) p_k(\lambda) - \eta_{k-1} p_{k-1}(\lambda), \quad k = 1, 2, \dots,$$

con condizioni iniziali $p_0 \equiv 0, p_1 \equiv 1$. Di conseguenza, dal Teorema di Favard (3) esiste una misura positiva per la quale questi polinomi sono ortogonali e vale il seguente Teorema, che lega l'Algoritmo di Lanczos con la teoria sviluppata nel Capitolo 1.

Teorema 14. Si considerino i vettori di Lanczos normalizzati v^k . Esiste una misura α tale che

$$\langle v^k, v^l \rangle = \langle p_k, p_l \rangle = \int_a^b p_k(\lambda) p_l(\lambda) d\alpha(\lambda),$$

con $a \leq \lambda_1$ e $b \geq \lambda_n$ dipendenti dagli autovalori di A .

Dimostrazione. Poiché la matrice A è simmetrica ammette decomposizione spettrale $A = Q\Lambda Q^T$, con Λ matrice diagonale degli autovalori denotati in ordine crescente:

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n.$$

Possiamo quindi definire $p_k(A) = Q p_k(\lambda) Q^T$ con

$$p_k(\lambda) = \begin{bmatrix} p_k(\lambda_1) & & & \\ & \ddots & & \\ & & \ddots & \\ & & & p_k(\lambda_n) \end{bmatrix},$$

e sfruttando l'ortonormalità dei v^j e il Teorema precedente si ha:

$$\begin{aligned} \langle v^k, v^l \rangle &= (v^1)^T p_k(A)^T p_l(A) v^1, \\ &= (v^1)^T Q p_k(\lambda)^T Q^T Q p_l(\lambda) Q^T v^1, \\ &= (v^1)^T Q p_k(\lambda)^T p_l(\lambda) Q^T v^1, \\ &= \sum_{j=1}^n p_k(\lambda_j) p_l(\lambda_j) [\hat{v}_j]^2, \end{aligned}$$

con l'uso della notazione $\hat{v} = Q^T v^1$.

L'ultima somma descrive un prodotto scalare discreto tra i polinomi p_k e p_l ; supponendo per semplicità che gli autovalori di A siano distinti può essere interpretata come un'integrale di Riemann-Stieltjes di una misura a gradini,

$$\alpha(\lambda) = \begin{cases} 0 & \lambda < \lambda_1, \\ \sum_{j=1}^i [\hat{v}_j]^2 & \lambda_i \leq \lambda < \lambda_{i+1}, \\ \sum_{j=1}^n [\hat{v}_j]^2 & \lambda_n \leq \lambda. \end{cases} \quad (28)$$

che per definizione è la misura richiesta dal Teorema. \square

L'Algoritmo di Lanczos fornisce dunque un esempio di polinomi ortonormali per una misura ignota (ovviamente non si procede a calcolare autovalori e autovettori di A che definiscono la misura) in quanto sono descritti dalla loro relazione di ricorrenza, riassunta dalla matrice J_m , definita a partire dal vettore iniziale v^1 . Dalla teoria, l'algoritmo costruisce una base ortogonale per il sottospazio di Krylov K_m ma nella pratica si perde ortogonalità a causa di una combinazione di errori di arrotondamento e altri derivanti dalla convergenza degli autovalori della matrice J_m a quelli della matrice A , che aumentano gli errori. Si deve dunque effettuare una reortogonalizzazione dei vettori v^m : questo può essere fatto ad ogni iterazione, e nel caso in cui risulti troppo costoso ci si può avvalere di tecniche specifiche sviluppate da Parlett in [17] e Simon in [18] o di modificare l'algoritmo in linea con le modifiche introdotte per l'Algoritmo di Arnoldi, come fatto da Saad in [14].

3.4 Algoritmo di Lanczos non simmetrico

L'Algoritmo di Lanczos non simmetrico costituisce un'estensione dell'Algoritmo 2 alle matrici non simmetriche che a differenza dell'Algoritmo di Arnoldi utilizza una matrice tridiagonale e conseguentemente genera una relazione di ricorrenza a tre termini tra i vettori costruiti. La struttura tridiagonale favorisce l'efficienza computazionale del metodo e consente anche in questo caso di costruire un legame con la teoria dei polinomi ortogonali.

L'Algoritmo di Lanczos non simmetrico può anche essere trovato in letteratura col nome di Algoritmo di biortogonalizzazione in quanto lavora su due spazi di Krylov differenti, ovvero

il consueto

$$K_m(A, v) = \text{span} \{ v, Av, \dots, A^{m-1}v \}$$

e il nuovo spazio

$$K_m(A^T, w) = \text{span} \{ w, A^T w, \dots, (A^T)^{m-1}w \},$$

con l'obbiettivo di creare due basi ortogonali, $\{ v^i \}_{i=1}^m$ per $K_m(A, v)$ e $\{ w^i \}_{i=1}^m$ per $K_m(A^T, w)$ che siano anche mutualmente ortogonali, ovvero con la proprietà

$$\langle v^i, w^j \rangle = \delta_i^j, \quad i, j = 1, \dots, m. \quad (29)$$

Essendo legate tra loro dalla relazione (29) le due basi devono essere costruite contemporaneamente, pertanto per descrivere il passo k per generare v^{k+1} e w^{k+1} servono due equazioni matriciali, ovvero

$$\begin{cases} AV_k = V_k T_k + \beta_{k+1} v^{k+1} (e^k)^T, \\ A^T W_k = W_k T_k^T + \gamma_{k+1} w^{k+1} (e^k)^T. \end{cases} \quad (30)$$

In questo caso T_k ha la consueta forma tridiagonale ma non è simmetrica:

$$T_k = \begin{bmatrix} \alpha_1 & \gamma_2 & & & \\ \beta_2 & \alpha_2 & \gamma_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{k-1} & \alpha_{k-1} & \gamma_k \\ & & & \beta_k & \alpha_k \end{bmatrix}.$$

Come fatto in precedenza nel caso degli algoritmi di Lanczos e Arnoldi, cerchiamo di ricavare v^{k+1} e w^{k+1} utilizzando le equazioni (30): svolgendo i calcoli si ottengono le due equazioni

$$\begin{cases} Av^k = \gamma_k v^{k-1} + \alpha_k v^k + \beta_{k+1} v^{k+1}, \\ A^T w^k = \beta_k w^{k-1} + \alpha_k w^k + \gamma_{k+1} w^{k+1}, \end{cases} \quad (31)$$

nella quale introduciamo la solita notazione per esplicitare i nuovi vettori, ottenendo

$$\begin{cases} \tilde{v}^k = \beta_{k+1} v^{k+1} = Av^k - \gamma_k v^{k-1} - \alpha_k v^k, \\ \tilde{w}^k = \gamma_{k+1} w^{k+1} = A^T w^k - \beta_k w^{k-1} - \alpha_k w^k, \end{cases} \quad (32)$$

con l'idea di poter calcolare i vettori utilizzando

$$\begin{cases} v^{k+1} = \frac{\tilde{v}^k}{\beta_{k+1}}, \\ w^{k+1} = \frac{\tilde{w}^k}{\gamma_{k+1}}. \end{cases}$$

Utilizzando ora la biortogonalità delle due famiglie per poter ricavare β_{k+1} e γ_{k+1} . Si ha

$$1 = \langle v^{k+1}, w^{k+1} \rangle = \frac{\langle \tilde{v}^k, \tilde{w}^k \rangle}{\beta_{k+1}\gamma_{k+1}},$$

che implica

$$\langle \tilde{v}^k, \tilde{w}^k \rangle = \beta_{k+1}\gamma_{k+1}. \quad (33)$$

L'equazione (33) non definisce i due coefficienti, ma permette varie scelte per gli stessi. Una possibilità è data dalla coppia

$$\begin{cases} \beta_{k+1} = |\langle \tilde{v}^k, \tilde{w}^k \rangle|^{1/2}, \\ \gamma_{k+1} = \text{sgn}(\langle \tilde{v}^k, \tilde{w}^k \rangle)\beta_{k+1}. \end{cases}$$

Per ricavare i coefficienti α_k riutilizziamo l'ortonormalità, moltiplicando le due equazioni (30) rispettivamente per w^k e v^k . Svolti i calcoli si ottiene

$$\begin{cases} \alpha_k = \langle Av^k, w^k \rangle = (w^k)^T Av^k, \\ \alpha_k = \langle A^T w^k, v^k \rangle = (v^k)^T A^T w^k = ((w^k)^T Av^k)^T, \end{cases}$$

ma le due espressioni sono uguali in quanto rappresentanti numeri reali e se ne può semplicemente scegliere una.

L'algoritmo può essere interpretato come un doppio metodo di proiezione: si ha infatti che T_m è la proiezione di A ottenuta da un metodo obliquo rispetto a $K_m(A, v)$ e ortogonale a $K_m(A^T, w)$, mentre T_m^T rappresenta la proiezione di A^T in maniera obliqua a $K_m(A^T, w)$ e ortogonale a $K_m(A, v)$.

I due operatori A, A^T assumono un ruolo duale e di fatto possono essere risolti in maniera parallela due sistemi lineari, uno dipendente da A e uno da A^T . Questo può risultare utile in applicazioni specifiche che si appoggiano ad una coppia di sistemi lineari, ad esempio la risoluzione di un problema di Programmazione Lineare tramite la dualità.

Algoritmo 3: Algoritmo di Lanczos non simmetrico

```
 $\beta_0 \gamma_0 = \langle v, w \rangle;$   
 $v^1 = v/\beta_0, \quad \gamma_1 \equiv 0, \quad v^0 \equiv 0;$   
 $w^1 = w/\gamma_0, \quad \beta_1 \equiv 0, \quad w^0 \equiv 0;$   
for  $j = 1, \dots, m$  do  
|  $\alpha_j = \langle Av^j, w^j \rangle;$   
|  $\tilde{v}^j = Av^j - \gamma_j v^{j-1} - \alpha_j v^j;$   
|  $\tilde{w}^j = A^T w^j - \beta_j w^{j-1} - \alpha_j w^j;$   
|  $\delta_j = |\langle \tilde{v}^j, \tilde{w}^j \rangle|;$   
|  $\beta_{j+1} = \sqrt{|\delta_j|};$   
|  $\gamma_{j+1} = \text{sgn}(\delta_j) \beta_{j+1};$   
| if  $\langle \tilde{v}^j, \tilde{w}^j \rangle = 0$  then  
| | stop (breakdown)  
| end  
| else  
| |  $v^{j+1} = \tilde{v}^j / \beta_{j+1};$   
| |  $w^{j+1} = \tilde{w}^j / \gamma_{j+1};$   
| end  
end
```

Figura 3: Algoritmo di Lanczos non simmetrico in aritmetica esatta, a partire da due vettori v e w .

L'analisi dei breakdown diventa molto più complessa se confrontata con i metodi di Arnoldi e Lanczos. Questi accadono quando $\langle \tilde{v}^j, \tilde{w}^j \rangle = 0$, e possono distinguersi in tre casistiche.

1. $\tilde{v}^j = 0$. Si è raggiunta la dimensione massima del sottospazio $K_m(A, v)$ e così come negli algoritmi 1 e 2 può essere accolto positivamente nella risoluzione del sistema lineare $Ax = b$. Il sottospazio di Krylov è invariante rispetto all'operatore A e dunque la soluzione approssimante data dal metodo di proiezione è la soluzione esatta del sistema.
2. $\tilde{w}^j = 0$. In questo caso è il sottospazio $K_m(A^T, w)$ ad essere invariante rispetto ad A^T e non si può dire nulla sulla soluzione del sistema lineare $Ax = b$. Se si sta utilizzando l'algoritmo per la risoluzione parallela di due sistemi si ha la soluzione esatta del sistema duale governato da A^T , e non sempre questo implica che si è raggiunta la soluzione del sistema primale (dipende dall'applicazione).
3. $\tilde{v}^j, \tilde{w}^j \neq 0$. Questa situazione è nota in letteratura come *serious breakdown* e si ha che nessuno dei due sottospazi è invariante, dunque non si è trovata alcuna soluzione esatta del sistema, ma l'algoritmo non può proseguire. Se si sta utilizzando l'algoritmo per stimare gli autovalori di A tramite la matrice T_m , il breakdown può essere aggirato semplicemente facendo ripartire l'algoritmo con diversi vettori iniziali. Nelle altre applicazioni dell'algoritmo questa soluzione non è utilizzabile e si può far ricorso a tecniche che cercano di evitare a priori il serious breakdown o di gestirlo e proseguire l'esecuzione in altro modo.

Da un punto di vista pratico questo algoritmo va confrontato con quello di Arnoldi, rispetto al quale si ha un notevole vantaggio dal punto di vista della memoria se utilizzato per stimare autovalori (in quanto non è necessario conservare tutta la matrice V_m). La matrice T_m sulla quale si lavora ha il vantaggio di essere in forma tridiagonale e dunque si possono utilizzare tecniche apposite sia per calcolarne gli autovalori che per risolvere sistemi lineari ad essa associata. Anche questo algoritmo, come i precedenti, soffre di perdita di ortogonalità: in questo caso si deve andare a correggere la biortogonalità (29) per contrastare gli errori di

calcolo generati dagli arrotondamenti con un procedimento specifico che generalizza quello di Gram-Schmidt alle due famiglie di vettori costruite.

4 Grafi

Un grafo *semplice* è una struttura discreta indicata con $G = (V, E)$, dove V è un insieme finito detto *insieme dei nodi* con cardinalità $|V|$ ed E è una relazione simmetrica e non riflessiva tra i nodi, rappresentante gli archi. Se la relazione E è non simmetrica il grafo è detto *orientato*. Questa definizione non consente la presenza di collegamenti multipli tra nodi o di auto-collegamenti tra un nodo e sé stesso. È possibile anche associare un peso ad ogni arco, ottenendo quindi un grafo *pesato*. Una sequenza di vertici v_1, \dots, v_N collegati tra loro da archi (dunque $(v_i, v_{i+1}) \in E$, $i = 1, \dots, N$) è detta *cammino* da v_1 a v_N di lunghezza N . Nel caso in cui $v_1 = v_N$ il cammino è detto *chiuso*.

La definizione di grafo *semplice* è sufficiente per gli sviluppi di questa tesi, ma si potrebbe estendere questa definizione per arrivare ai grafi *complessi*, che consentono di descrivere molti dettagli di una rete fisica e costituiscono il punto di una partenza per una teoria dalle molteplici applicazioni nel mondo moderno in ambito matematico, informatico, finanziario, biologico *et cetera* descritta in [19].

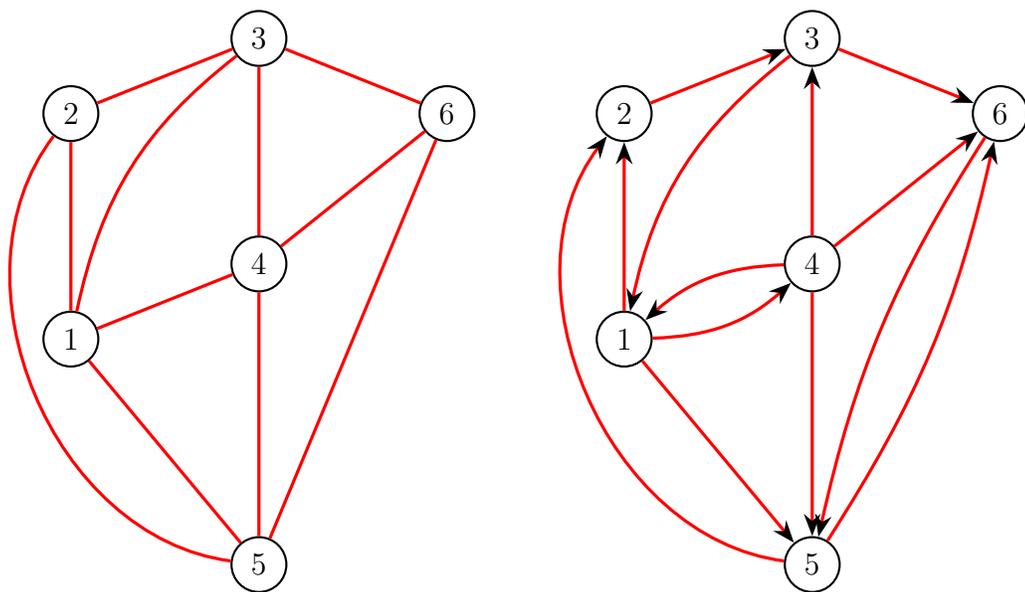
Per poter rappresentare un grafo in maniera sintetica si definisce una matrice A di dimensione $|V| \times |V|$ detta *di adiacenza* definita come

$$A_{ij} = \begin{cases} 1 & \text{se } (v_i, v_j) \in E, \\ 0 & \text{altrimenti.} \end{cases}$$

Nella pagina seguente sono riportati due grafi, uno orientato e uno no, con le relative matrici di adiacenza. Si noti come soltanto nel caso del grafo non orientato la matrice risulti simmetrica. Definire la matrice di adiacenza consente di effettuare analisi del grafo basate sull'algebra lineare.

4.1 Comunicabilità

Nella maggior parte delle reti reali è interessante studiare la comunicazione che avviene, o può avvenire, tra i nodi: si lavora sotto l'assunzione che questa si manifesti lungo tutti i cammini possibili tra i nodi, e ovviamente i cammini più corti devono avere maggiore impatto nell'analisi. In generale trovare tutti i cammini tra due nodi è un problema complesso,



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

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

$$A^2 = \begin{bmatrix} 4 & 2 & 2 & 2 & 2 & 3 \\ 2 & 3 & 1 & 3 & 1 & 2 \\ 2 & 1 & 4 & 2 & 4 & 1 \\ 2 & 3 & 2 & 4 & 2 & 2 \\ 2 & 1 & 4 & 2 & 4 & 1 \\ 3 & 2 & 1 & 2 & 1 & 3 \end{bmatrix}$$

$$B^2 = \begin{bmatrix} 1 & 1 & 2 & 0 & 1 & 2 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 2 & 0 \\ 1 & 2 & 0 & 1 & 2 & 2 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 4: Esempi di grafo non orientato e orientato con relative matrici di adiacenza, rappresentanti i cammini di lunghezza 1 e 2.

ma per questa analisi è sufficiente sapere quanti sono, anche senza conoscere i vertici che li compongono. Fortunatamente esiste un procedimento molto semplice per conoscere il numero di cammini tra due nodi i e j di lunghezza k fissata. È sufficiente infatti calcolare la matrice A^k e considerare l'entrata $(A^k)_{ij}$. Nella Figura 4 vengono riportate, oltre alle matrici di adiacenza, anche le matrici A^2, B^2 descriventi i cammini di lunghezza 2. Per stimare la comunicabilità tra i nodi si può pensare di sommare tutte le matrici A^k opportunamente pesate. Una scelta naturale consiste nell'usare $\frac{1}{k!}$ come peso applicato ad A^k in modo da penalizzare i cammini lunghi, arrivando ad estendere la funzione esponenziale alle matrici:

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!}. \quad (34)$$

Si fa subito notare come un elemento della matrice e^A possa essere espresso tramite una forma bilineare

$$(e^A)_{ij} = (e^i)^T e^A e^j, \quad (35)$$

dove e^i, e^j sono l' i -esimo e il j -esimo vettore colonna della matrice identità di ordine $|V|$, numero di nodi del grafo analizzato.

4.2 Misure di centralità

Uno degli obiettivi principali nell'analisi di una rete rappresentata da un grafo è individuare i nodi più *importanti*. Per dare un senso all'aggettivo si introducono le misure di centralità, con l'obiettivo per quantificare l'importanza di un nodo. Le più semplici misure di centralità sono i *gradi* del nodo, dipendenti unicamente dai collegamenti da e per il nodo stesso. Il grado *di ingresso* di un nodo viene denotato con k_i^{in} ed è definito come il numero di archi entranti nel nodo, mentre il grado *di uscita* k_i^{out} rappresenta il numero di archi uscenti. Possono essere calcolati in maniera efficiente tramite la matrice di adiacenza:

$$\begin{cases} k_i^{\text{in}} = (u^T A)_i, \\ k_i^{\text{out}} = (A u)_i, \\ u = \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}^T. \end{cases}$$

Si noti come questi indici coincidano per un grafo non orientato e possano essere calcolati (in maniera non efficiente) utilizzando la matrice di adiacenza al quadrato:

$$k_i^{\text{in}} = k_i^{\text{out}} = k_i = (A^2)_{ii}.$$

Volendo avere una misura più rappresentativa, possiamo ripetere il ragionamento svolto in precedenza per arrivare a definire la *subgraph centrality* [20], che tiene conto della capacità del nodo di far parte di cammini chiusi. Il numero di cammini chiusi di lunghezza k può essere espresso come $(A^k)_{ii}$. Per considerarli tutti si può definire in generale

$$f_i(A) = \left(\sum_{k=0}^{\infty} c_k A^k \right)_{ii},$$

dove i pesi c_k sono definiti in modo da rendere convergente la serie. Possiamo nuovamente sceglierli in modo da arrivare alla funzione esponenziale e definire la *subgraph centrality* più comunemente usata

$$f_i(A) = \left(\sum_{k=0}^{\infty} \frac{A^k}{k!} \right)_{ii} = (e^A)_{ii}. \quad (36)$$

Anche in questo caso possiamo esprimere il calcolo tramite una forma bilineare:

$$(e^A)_{ii} = (e^i)^T e^A e^i. \quad (37)$$

5 Funzioni Matriciali

Alla luce di quanto esposto nel Capitolo 4, nell'ambito dell'analisi di una rete a larga scala c'è la necessità di calcolare forme bilineari associate a matrici di grandi dimensioni: supponiamo dunque di voler calcolare la *subgraph centrality* per un nodo i di un grafo, ovvero valutare la forma bilineare

$$(e^i)^T e^A e^i,$$

5.1 Utilizzo dell'algoritmo di Lanczos

Affrontiamo il problema con una notazione più generale, ovvero cercando di valutare la forma bilineare

$$u^T f(A) u, \tag{38}$$

dove u è un vettore generico e f è una funzione matriciale, ovvero si ha

$$f(A) = \sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!} A^k, \quad \text{con } f \text{ di classe } C^\infty.$$

Assumendo che A sia una matrice simmetrica, e dunque unitariamente diagonalizzabile, la (38) diventa una forma quadratica e possiamo utilizzare la decomposizione spettrale

$$A = Q \Lambda Q^T, \tag{39}$$

dove Q è la matrice degli autovettori normalizzati e Λ è la matrice diagonale degli autovalori ordinati in senso non decrescente

$$\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n.$$

Per le matrici diagonalizzabili vale la formula (si rimanda a [21] per dettagli)

$$f(A) = Q f(\Lambda) Q^T,$$

che possiamo utilizzare in (38) ottenendo

$$\begin{aligned} u^T f(A) u &= u^T Q f(\Lambda) Q^T u, \\ &= \hat{u}^T f(\Lambda) \hat{u}, \\ &= \sum_{i=1}^n f(\lambda_i) [\hat{u}_i]^2. \end{aligned}$$

L'ultima sommatoria può essere interpretata come un integrale di Riemann-Stieltjes:

$$I(f) = u^T f(A)u = \int_a^b f(\lambda) d\alpha(\lambda),$$

dove la misura α è a gradini e definita da:

$$\alpha(\lambda) = \begin{cases} 0 & \lambda < \lambda_1, \\ \sum_{j=1}^i [\hat{u}_j]^2 & \lambda_i \leq \lambda < \lambda_{i+1}, \\ \sum_{j=1}^n [\hat{u}_j]^2 & \lambda_n \leq \lambda. \end{cases} \quad (40)$$

Questa misura è esattamente la misura (28) descritta dal metodo di Lanczos applicato alla matrice A con vettore iniziale u . Di conseguenza possiamo stimare l'integrale, e dunque la forma quadratica, con una formula di quadratura basata sull'algoritmo di Lanczos avvalendoci della teoria sviluppata nei capitoli 1 e 2. Si pone infatti

$$u^T f(A)u \approx \mathcal{G}_m f,$$

con

$$\mathcal{G}_m f = \sum_{j=1}^m f(\mu_j)\omega_j,$$

dove i μ_j sono gli autovalori della matrice di Jacobi J_m ottenuta dopo m iterazioni dell'algoritmo di Lanczos (con vettore iniziale u) e i pesi ω_j sono le prime componenti al quadrato dei corrispondenti autovettori.

5.2 Utilizzo dell'algoritmo di Lanczos non simmetrico

Vediamo ora come l'algoritmo di Lanczos non simmetrico possa essere utilizzato per stimare forme bilineari derivanti da funzioni matriciali, facendo la sola assunzione che la matrice A sia diagonalizzabile. Questo approccio consente di calcolare efficientemente la *subgraph centrality* di un nodo in un grafo orientato, avente matrice di adiacenza non simmetrica. Seguendo la trattazione di [22], trattiamo il caso più generale possibile, ovvero il calcolo della forma bilineare

$$u^T f(A)u', \quad (41)$$

dove u e u' sono vettori generici, A è supposta diagonalizzabile ed f è di classe C^∞ . Per prima cosa utilizziamo la fattorizzazione spettrale della matrice per esprimere la forma bilineare in forma di integrale di Stieltjes come fatto in precedenza ottenendo

$$u^T f(A) u' = \underbrace{u^T S}_\mu f(\Lambda) \underbrace{S^{-1} u'}_{\mu'} = \sum_{j=1}^N f(\lambda_j) \mu_j \mu'_j = \int f d\alpha = \mathcal{I}(f),$$

dove la misura α è relativa a un insieme nel piano complesso contenente gli autovalori di A . Si vuole quindi stimare $\mathcal{I}(f)$ tramite la formula di quadratura gaussiana avente la forma

$$\mathcal{G}_m f = \sum_{k=1}^m f(z_k) \omega_k, \quad (42)$$

dove gli z_k sono zeri di un polinomio ortogonale di grado k rispetto alla forma bilineare

$$[f, g] = \mathcal{I}(fg) = \int fg d\alpha, \quad (43)$$

definita tramite la misura α e i pesi ω_k sono tali da rendere la formula (42) interpolatoria, ossia esatta per un qualsiasi polinomio di grado minore o uguale a $m - 1$. Supponendo per semplificare la notazione di avere $\langle u, u' \rangle = 1$ (se ciò non fosse vero è possibile riscalarne i vettori e tenerne conto alla fine del procedimento) possiamo utilizzare l'algoritmo di Lanczos non simmetrico con vettori iniziali

$$v^1 = u', \quad w^1 = u,$$

per costruire due famiglie di vettori biortogonali, aventi quindi la proprietà

$$\langle v^k, w^j \rangle = \delta_k^j,$$

che possono essere scritte come

$$v^j = p_{j-1}(A)v^1, \quad w^j = q_{j-1}(A^T)w^1. \quad (44)$$

I polinomi p e q non vengono esplicitamente calcolati dall'algoritmo, ma possiamo facilmente dimostrare che questi sono biortogonali rispetto alla forma bilineare (43). Si ha infatti

$$\begin{aligned} [q_k, p_j] &= \int q_k p_j d\alpha = u^T S q_k(\Lambda) p_j(\Lambda) S^{-1} u' = \\ &= u^T S q_k(\Lambda) S^{-1} S p_j(\Lambda) S^{-1} u' = u^T q_k(A) p_j(A) u' = \\ &= \langle w^{k+1}, v^{j+1} \rangle = \delta_k^j. \end{aligned}$$

Utilizzando le equazioni matriciali (30) che descrivono l'algoritmo, si possono dimostrare due relazioni di ricorrenza per i polinomi p e q che consentono di scrivere due nuove equazioni

$$\begin{bmatrix} p_0(z) & \cdots & p_{m-1}(z) \end{bmatrix} T_m = z \begin{bmatrix} p_0(z) & \cdots & p_{m-1}(z) \end{bmatrix} - \beta_m \begin{bmatrix} 0 & \cdots & 0 & p_m(z) \end{bmatrix}, \quad (45)$$

$$\begin{bmatrix} q_0(z) & \cdots & q_{m-1}(z) \end{bmatrix} T_m^T = z \begin{bmatrix} q_0(z) & \cdots & q_{m-1}(z) \end{bmatrix} - \gamma_m \begin{bmatrix} 0 & \cdots & 0 & q_m(z) \end{bmatrix}, \quad (46)$$

che mostrano come gli zeri di $p_m(z)$ siano gli autovalori di T_m e quelli di $q_m(z)$ siano autovalori di T_m^T . Gli autovalori di T_m coincidono con quelli di T_m^T e li denotiamo con z_1, \dots, z_m .

Introducendo le tre matrici

$$P_m = \begin{bmatrix} p_0(z_1) & \cdots & p_{m-1}(z_1) \\ \vdots & \ddots & \vdots \\ p_0(z_m) & \cdots & p_{m-1}(z_m) \end{bmatrix}, \quad Q_m = \begin{bmatrix} q_0(z_1) & \cdots & q_{m-1}(z_1) \\ \vdots & \ddots & \vdots \\ q_0(z_m) & \cdots & q_{m-1}(z_m) \end{bmatrix},$$

$$D_m = \text{diag} [z_1 \cdots z_m],$$

dall'equazione (45) ricaviamo

$$P_m T_m = D_m P_m \implies T_m P_m^{-1} = P_m^{-1} D_m,$$

e in maniera simile dall'equazione (46) si ottiene

$$Q_m T_m^T = D_m Q_m \implies T_m Q_m^T = Q_m^T D_m.$$

Le matrici P_m^{-1} e Q_m^T sono entrambe costituite da autovettori di T_m , dunque

$$\exists \Omega_m = \text{diag} [\omega_1 \cdots \omega_m] \quad \text{tale che} \quad P_m^{-1} = Q_m^T \Omega_m. \quad (47)$$

Dal momento che i polinomi sono definiti da (44) si ha $q_0(z) \equiv 1$ e la prima riga di Q_m^T è formata da soli 1. Dalla (47) si ricava allora

$$\omega_k = (P_m^{-1})_{1,k},$$

che, si dimostrerà in seguito, sono i pesi della formula di quadratura. Sempre dall'equazione (47) si ottengono la formula

$$P_m^{-1} = Q_m^T \Omega_m \implies \Omega_m^{-1} = P_m Q_m^T \implies \omega_k = \left(\sum_{j=0}^{m-1} p_j(z_k) q_j(z_k) \right)^{-1}, \quad (48)$$

che esplicita i valori dei coefficienti ω_k , e la proprietà

$$P_m^{-1} = Q_m^T \Omega_m \implies Q_m^T \Omega_m P_m = I_m \implies \sum_{k=1}^m q_j(z_k) p_i(z_k) \omega_k = \delta_i^j. \quad (49)$$

Possiamo ora dimostrare il seguente teorema.

Teorema 15. *La formula di quadratura (42), con le scelte appena descritte per i nodi z_k e i pesi w_k , è esatta per un qualsiasi polinomio f di grado minore o uguale a $2m - 1$.*

Dimostrazione. Sia

$$f \in \mathbb{P}_{2m-1} \implies f(z) = p_m(z)s(z) + r(z),$$

con

$$r(z) = \sum_{j=0}^{m-1} r_j q_j(z), \quad s(z) = \sum_{j=0}^{m-1} s_j q_j(z).$$

Utilizzando la sua fattorizzazione, integriamo f ottenendo

$$\mathcal{I}f = \int f d\alpha = [p_m, s] + [r, 1] = [r, 1] = [r, q_0] = r_0.$$

Applicando invece la formula di quadratura si ha

$$\mathcal{G}_m f = \mathcal{G}_m(p_m s) + \mathcal{G}_m r = \mathcal{G}_m r,$$

dal momento che $\mathcal{G}_m(p_m s) = 0$ in quanto i nodi utilizzati sono gli zeri di p_m . Utilizzando la proprietà (49), derivante dalla scelta dei pesi, si ricava

$$\begin{aligned} \mathcal{G}_m f &= \mathcal{G}_m r = \\ &= \sum_{k=1}^m r(z_k) \omega_k = \\ &= \sum_{k=1}^m \left(\sum_{j=0}^{m-1} r_j q_j(z_k) \right) \omega_k = \\ &= \sum_{k=1}^m \left(\sum_{j=0}^{m-1} r_j q_j(z_k) \right) \omega_k \underbrace{p_0(z_k)}_{=1} = \\ &= \sum_{j=0}^{m-1} r_j \underbrace{\left(\sum_{k=1}^m q_j(z_k) p_0(z_k) \omega_k \right)}_{=\delta_0^j} = \\ &= r_0. \end{aligned}$$

Confrontando le due espressioni si ha $\mathcal{I}f = \mathcal{G}_m f$. □

Considerando una qualsiasi fattorizzazione spettrale di T_m

$$T_m = Z_m D_m Z_m^{-1},$$

possiamo definire i vettori

$$\left[\tilde{\omega}_1 \cdots \tilde{\omega}_m \right] = (e^1)^T Z_m, \quad \left[\hat{\omega}_1 \cdots \hat{\omega}_m \right] = Z^{-1} e^1,$$

e dal momento che i prodotti $\omega_k = \hat{\omega}_k \tilde{\omega}_k$ non dipendono dalla fattorizzazione scelta, gli ω_k sono proprio quelli definiti nel corso della dimostrazione e si può scrivere

$$\begin{aligned} \mathcal{G}_m f &= \sum_{k=1}^m f(z_k) \omega_k = \\ &= \sum_{k=1}^m \hat{\omega}_k f(z_k) \tilde{\omega}_k = \\ &= (e^1)^T Z_m f(D_m) Z^{-1} e^1 = \\ &= (e^1)^T f(T_m) e^1. \end{aligned}$$

che fornisce una scrittura più compatta per la formula di quadratura permettendo di scrivere semplicemente

$$\mathcal{G}_m f = (e^1)^T f(T_m) e^1, \tag{50}$$

a meno di fattori di scala derivanti dall'aver inizialmente imposto $\langle u, u' \rangle = 1$.

6 Risultati

Questo capitolo riporta i risultati ottenuti dall'algoritmo descritto nella sezione 5.2 ai fini del calcolo della forma bilineare (41), con particolare attenzione al calcolo degli indici di Estrada su reti complesse.

Tutti i risultati sono stati ottenuti su un laptop dotato di CPU quad-core Intel i7-4710HQ e 8 GB di memoria RAM a disposizione, utilizzando il software MATLAB. L'implementazione si basa sull'algoritmo 3 con in aggiunta i seguenti tre accorgimenti di carattere implementativo.

- Una funzione opzionale di reortogonalizzazione delle due famiglie, che può essere chiamata ad ogni iterazione, al fine di assicurare la proprietà $\langle v^k, w^j \rangle = \delta_k^j$ che tende ad essere persa a causa dell'accumularsi di errori di arrotondamento.
- Shift degli autovalori della matrice di adiacenza A : per evitare overflow durante l'esecuzione, si lavora sulla matrice $A - \mu I$, dove μ è una stima del massimo modulo degli autovalori di A . Questo procedimento porta al calcolo di una formula di quadratura \mathcal{G}'_m diversa da (50), con la proprietà $\mathcal{G}_m = e^\mu \mathcal{G}'_m$. Dopo aver calcolato \mathcal{G}'_m si può provare ad effettuare la moltiplicazione per e^μ per ottenere \mathcal{G}_m . Se quest'ultimo passaggio genera overflow, si utilizza \mathcal{G}'_m come risultato finale e il valore di μ viene conservato per indicare il fattore di scala mancante.
- Criterio di stop multiplo che tiene conto: del verificarsi di un breakdown, del raggiungimento di un numero massimo di iterazioni prefissato o della convergenza della formula di quadratura. Quest'ultima viene raggiunta quando è verificata la condizione

$$\frac{|\mathcal{G}_m - \mathcal{G}_{m-1}|}{|\mathcal{G}_m|} \leq \tau.$$

Nei test effettuati si è posto $\tau = 10^{-3}$.

L'algoritmo è stato utilizzato per reti orientate e non orientate, anche se nel secondo caso si sarebbe potuta utilizzare la sua versione simmetrica, che avrebbe prodotto gli stessi risultati gestendo una sola famiglia di vettori al posto di due identiche, con conseguente miglioramento dal punto di vista prestazionale.

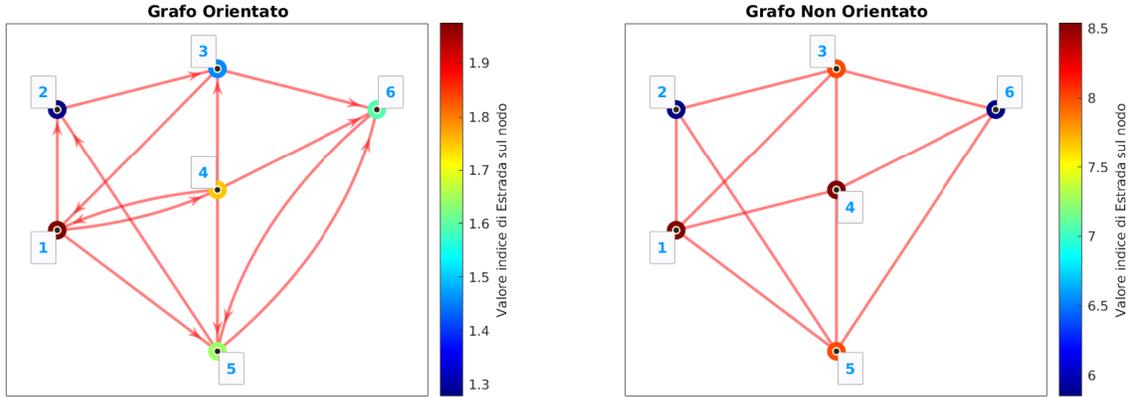


Figura 5: Analisi delle due reti d'esempio introdotte nel Capitolo 4

Al fine di comprendere meglio i risultati successivi, si riportano in figura 5 i valori dell'indice di Estrada per i nodi dei due grafi d'esempio introdotti in precedenza. Dalla figura si può notare, soprattutto nel caso del grafo orientato, come tale indice descriva in senso qualitativo la capacità di un nodo di raggiungere gli altri nodi della rete.

Si può costruire facilmente una rete con un nodo che causi una delle situazioni di breakdown elencate nella sezione 3.4: negli esempi proposti è sufficiente inserire un nodo aggiuntivo che conduca ad una matrice di adiacenza con una riga o una colonna composta da soli zeri. Nel caso delle reti orientate significa avere un nodo di solo arrivo o di sola partenza, mentre nel caso di reti non orientate si tratta di un nodo *isolato*. Nella figura 6 è riportata la rete ottenuta aggiungendo un nodo di solo arrivo (dunque con grado d'uscita pari a 0) alla rete orientata che si è usata in precedenza, insieme alla matrice di adiacenza A che si ottiene. In generale un nodo di indice l con questa conformazione porta ad avere $Ae^l = 0$ o $A^T e^l = 0$, portando ad un breakdown di primo o di secondo tipo alla prima iterazione. In questi casi si ha $A^k e^l = 0$ oppure $(e^l)^T A^k = 0$ per ogni $k = 1, \dots, m$, portando ad avere $(e^l)^T e^A e^l = 1$ e dunque il breakdown non è preoccupante perché l'indice di Estrada è noto dalla teoria. Si possono avere breakdown anche in situazioni in cui un nodo non sia di sola entrata o sola uscita, come nel caso dei *serious* breakdown, per i quali non c'è nulla da fare e si deve ricorrere a tecniche apposite per provare a continuare l'esecuzione e aggirarlo o per prevenirlo. In [6] viene descritta una generalizzazione dell'algoritmo che si basa sui polinomi ortogonali matriciali, che consentono il calcolo simultaneo di più indici di Estrada tramite

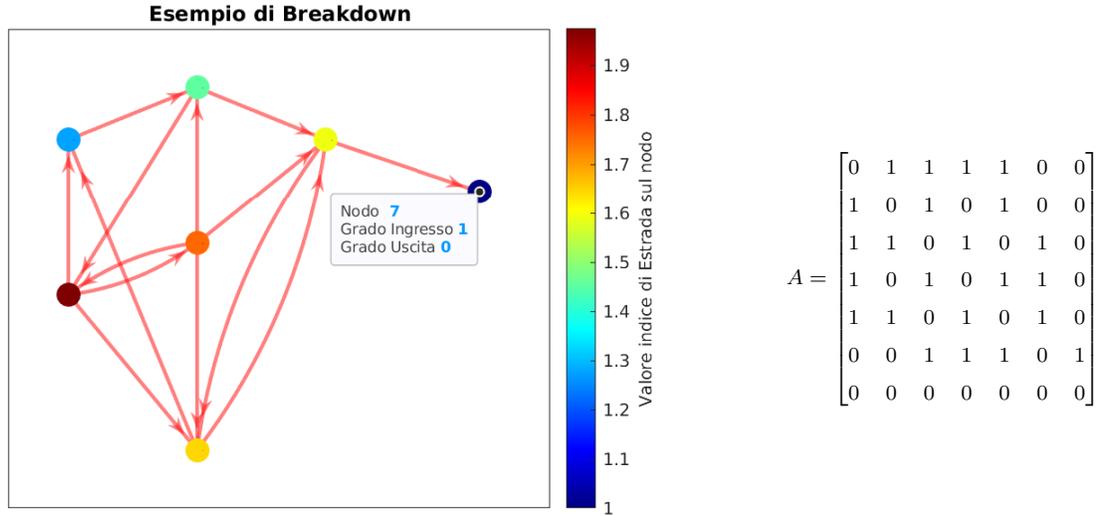


Figura 6: Il calcolo dell'indice di Estrada per il nodo 7 causa breakdown alla prima iterazione.

blocchizzazione dell'algoritmo: tale procedura può aiutare a prevenire i breakdown grazie all'inserimento di vettori di supporto *pieni*.

I breakdown iniziali sono di fatto causati dalla sparsità delle matrici di adiacenza utilizzate: il calcolo dell'indice di Estrada inizializza l'algoritmo di Lanczos con due vettori aventi una sola componente non nulla, che contribuiscono a generare vettori sparsi tendenti a risultare ortogonali tra loro.

La figura 7 mostra i risultati di un test effettuato su delle reti di 3000 nodi generati in maniera randomica facendo variare la densità di elementi non nulli nella matrice di adiacenza. La probabilità di breakdown è stimata sul calcolo dell'indice di Estrada in 300 nodi della rete. Dal primo grafico si nota che, come previsto, i breakdown diventano meno probabili all'aumentare degli elementi non nulli, seguendo un andamento lineare. La crescita del numero medio di iterazioni effettuate è da attribuire unicamente al mancato verificarsi dei breakdown, mentre quella del tempo medio di esecuzione dipende anche dal maggior numero di moltiplicazioni da effettuare per un prodotto matrice vettore su una matrice più densa.

Le tabelle 2 e 3 a fine capitolo riportano i risultati dell'algoritmo su due collezioni di matrici di adiacenza relative a reti reali rappresentanti varie forme di comunicazione. Le matrici di adiacenza rappresentano reti sociali (scambio di mail, collaborazioni scientifiche

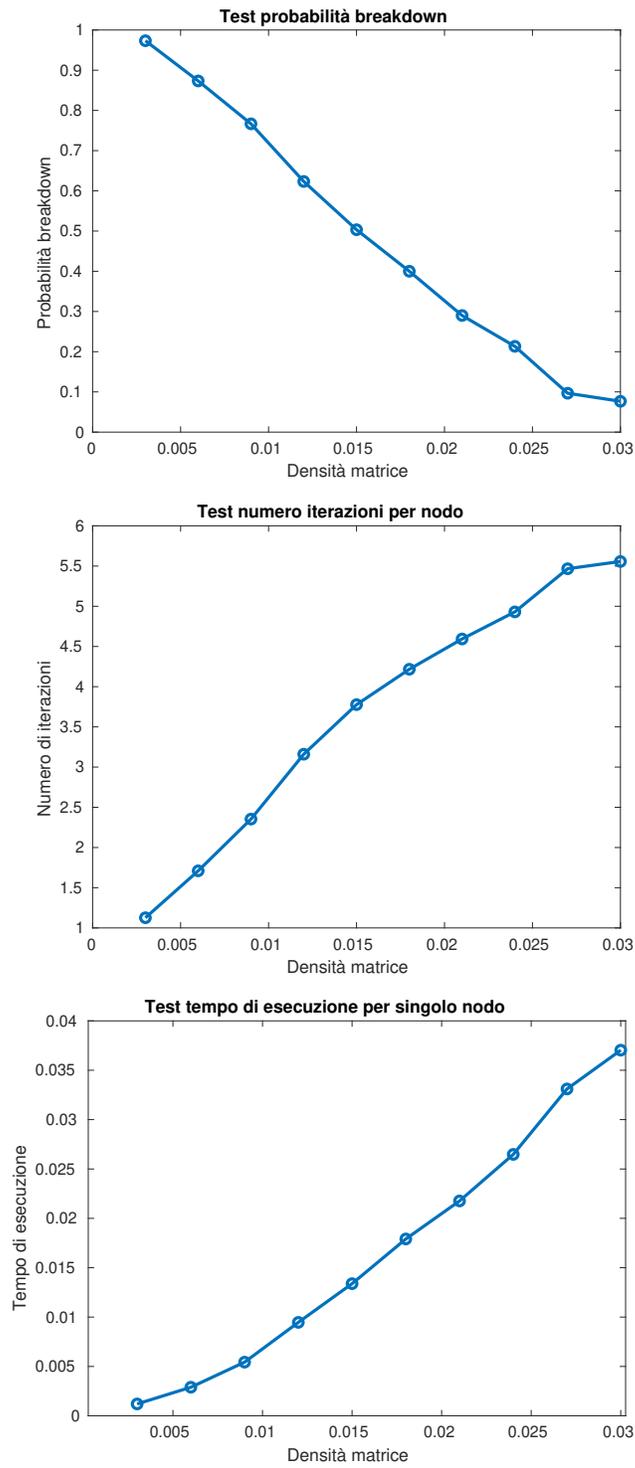


Figura 7: Analisi delle prestazioni al variare della densità.

tra ricercatori), biologiche (strutture proteiche o di metabolismi), di trasporto (collegamenti aerei, autostrade o trasporti pubblici), telematiche (porzioni di Facebook, Twitter e Wikipedia). I dati inseriti confrontano le prestazioni dell'algoritmo presentato con la funzione `expm` di `MATLAB` che calcola interamente l'esponenziale di matrice: oltre una certa dimensione tale confronto non è più praticabile per problemi legati alla saturazione della memoria RAM del computer utilizzato durante l'esecuzione della funzione. Per ogni matrice sono stati stimati i valori di 50 indici di Estrada al fine di confrontarli con i valori di `expm`, assunti come esatti. Si riportano nelle tabelle il numero di nodi e di archi per osservare come i tempi di calcolo e i breakdown siano nuovamente influenzati dalla sparsità delle matrici. Il CPU-time riportato per l'algoritmo di Lanczos corrisponde a quello necessario all'esecuzione sequenziale su tutti 50 nodi; si riporta anche il numero di iterazioni medie necessarie per la convergenza, il numero di breakdown incontrati e l'errore relativo più grande commesso tra i nodi di ogni singola matrice. Tra le reti orientate vi è quella del CTM, azienda di trasporto della città di Cagliari, rappresentante le fermate della rete di trasporto urbano cittadina, dove gli archi sono dati dai collegamenti tra le fermate a chiamata, percorsi dalle linee operative dell'azienda. Sfortunatamente tale matrice si è dimostrata troppo sparsa per essere analizzata tramite formula di quadratura.

Matrici	Nodi	Archi	CPU-Time (s)		Iterazioni	Breakdowns	Errore
			expm	Lanczos			
Mail	1133	10902	0.31	0.27	9.14	0	4.05 e-05
Autobahn	1168	2486	0.35	0.06	4.60	0	1.79 e-05
Yeast	2114	4480	1.96	0.15	4.60	17	2.83 e-05
Power	4941	13188	24.22	0.35	5.22	0	1.61 e-05
Collab	40421	351384	-	11.24	11.21	2	-
Facebook	63731	1634180	-	51.24	12.84	0	-

Tabella 2: Test sul calcolo di 50 indici di Estrada in reti non orientate.

Matrici	Nodi	Archi	CPU-Time (s)		Iterazioni	Breakdowns	Errore
			expm	Lanczos			
Airlines	235	2101	0.10	0.11	7.06	0	1.72 e-05
Celegans	306	2345	0.08	0.08	4.44	24	2.83 e-04
USA97	332	2126	0.05	-	-	50	-
Air500	500	24009	0.10	0.38	8.68	0	1.22 e-05
CTM	970	1238	0.13	-	-	50	-
Twitter	3556	188712	222.00	2.28	8.64	5	6.00 e-03
WikiVote	8297	103690	-	0.63	3.90	32	-
PGP	10680	24316	-	-	-	50	-
Wiki	49728	941425	-	23.41	10.96	6	-
Poisson	85623	2374949	-	13.72	4.44	0	-

Tabella 3: Test sul calcolo di 50 indici di Estrada in reti orientate.

Bibliografia

- [1] Gene Howard Golub e Gerard A. Meurant. *Matrices, moments and quadrature with applications*. Princeton University Press, 2010.
- [2] J. Favard. «Sur les polynomes de Tchebicheff.» French. In: *C. R. Acad. Sci., Paris* 200 (1935), pp. 2052–2053. ISSN: 0001-4036.
- [3] Francisco Marcellán e Renato Álvarez-Nodarse. «On the “Favard theorem” and its extensions». In: *Journal of Computational and Applied Mathematics* 127.1 (2001). Numerical Analysis 2000. Vol. V: Quadrature and Orthogonal Polynomials, pp. 231–254. ISSN: 0377-0427. DOI: [https://doi.org/10.1016/S0377-0427\(00\)00497-0](https://doi.org/10.1016/S0377-0427(00)00497-0). URL: <http://www.sciencedirect.com/science/article/pii/S0377042700004970>.
- [4] G. Szegő. *Orthogonal Polynomials*. American Mathematical Society colloquium publications v. 23. American Mathematical Society, 1959. ISBN: 9780821889527. URL: <https://books.google.it/books?id=3hcW8HBh7gsC>.
- [5] George W Struble. «Orthogonal polynomials: variable-signed weight functions». In: *Numerische Mathematik* 5.1 (1963), pp. 88–94.
- [6] C. Fenu, D. Martin, L. Reichel e G. Rodriguez. «Block Gauss and Anti-Gauss Quadrature with Application to Networks». In: *SIAM Journal on Matrix Analysis and Applications* 34.4 (2013), pp. 1655–1684. DOI: 10.1137/120886261.
- [7] C. Fenu, D. Martin, L. Reichel e G. Rodriguez. «Network Analysis via Partial Spectral Factorization and Gauss Quadrature». In: *SIAM Journal on Scientific Computing* 35.4 (2013). DOI: 10.1137/130911123.
- [8] Walter Gautschi. *Orthogonal Polynomials: Computation and Approximation*. Oxford University Press, 2004.
- [9] Carl Friedrich Gauss. *Methodus nova integralium valores per approximationem inveniendi*. apvd Henricvm Dieterich, 1815.
- [10] J. Stoer, R. Bartels, W. Gautschi, R. Bulirsch e C. Witzgall. *Introduction to Numerical Analysis*. Texts in Applied Mathematics. Springer New York, 2013. ISBN: 9780387217383. URL: <https://books.google.it/books?id=fh3vBwAAQBAJ>.

- [11] Herbert S Wilf. *Mathematics for the physical sciences*. Courier Corporation, 2013.
- [12] Gene H Golub e John H Welsch. «Calculation of Gauss quadrature rules». In: *Mathematics of computation* 23.106 (1969), pp. 221–230.
- [13] Gene H Golub. «Some modified matrix eigenvalue problems». In: *Siam Review* 15.2 (1973), pp. 318–334.
- [14] Yousef Saad. *Iterative methods for sparse linear systems*. Vol. 82. siam, 2003.
- [15] Walter Edwin Arnoldi. «The principle of minimized iterations in the solution of the matrix eigenvalue problem». In: *Quarterly of applied mathematics* 9.1 (1951), pp. 17–29.
- [16] Cornelius Lanczos. «An iteration method for the solution of the eigenvalue problem of linear differential and integral operators». In: *Journal of Research of the National Bureau of Standards* (1950).
- [17] Beresford N Parlett e David S Scott. «The Lanczos algorithm with selective orthogonalization». In: *Mathematics of computation* 33.145 (1979), pp. 217–238.
- [18] Horst D Simon. «The Lanczos algorithm with partial reorthogonalization». In: *Mathematics of computation* 42.165 (1984), pp. 115–142.
- [19] Ernesto Estrada. *The structure of complex networks: theory and applications*. Oxford University Press, 2016.
- [20] Ernesto Estrada e Juan A Rodriguez-Velazquez. «Subgraph centrality in complex networks». In: *Physical Review E* 71.5 (2005), p. 056103.
- [21] Nicholas J Higham. *Functions of matrices: theory and computation*. Vol. 104. Siam, 2008.
- [22] Daniela Calvetti, Lothar Reichel e Fiorella Sgallari. «Applications of anti-Gauss quadrature rules in linear algebra». In: *Applications and Computation of Orthogonal Polynomials*. Springer, 1999, pp. 41–56.